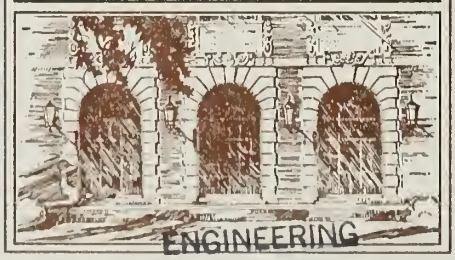


Center for Advanced Computation

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

URBANA, ILLINOIS 61801

CAC Document No. 26

AN INTRODUCTION TO THE ANALYSIS OF TIME SERIES

By

K. Miura

November 8, 1971

CAC Document No. 26

AN INTRODUCTION TO THE ANALYSIS OF TIME SERIES

by

K. Miura

Center for Advanced Computation
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

November 8, 1971

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the U.S. Army Research Office-Durham under Contract No. DAHC04 72-C-0001.

ABSTRACT

This document is an introduction to the analysis of time series using autocorrelations. The autoregressive, the moving average, and the mixed models for stationary and nonstationary time series -- are introduced. Several stages to analyse given time series are discussed. These are: identification, preestimation of parameters, the maximum likelihood estimation and diagnostic checking.

Numerical techniques are described in detail together with some examples.

TABLE OF CONTENTS

Chapter 1 Introduction

This document has been prepared as an introduction to the analysis of time series - or more precisely, estimation of parameters in discrete time series. A time series is a set of observations generated sequentially in time. There are continuous time series $Z(t)$ and discrete time series $\{Z_t\}$, but we are interested throughout this document only in discrete time series observed at equal interval. Data are measured at τ_0, τ_0+h, τ_0+2h, ····and they are denoted by, for example, $\{Z_1, Z_2, Z_3, \ldots\}$.

Analysis of time series is particularly important in fields where control and forecast are involved; observed data are inevitably contaminated by random noise. Take the stock market, for example. The offer by sellers will be affected by the previous price and the most recent change of price. Furthermore we must assume a term which is unpredictable. Therefore the price X_t can be described as

$$X_t = \alpha X_{t-1} + \beta(X_{t-1} - X_{t-2}) + a_t$$

This is a very simple model and is a linear combination of past values and random term. It is called auto-regressive process. After defining a model, we can estimate α and β from a given set of observations $\{X_1, X_2 \ldots.\}$. Then we will be able to investigate the effect of the previous price or change of prices to the current price of the stock and also be able to predict the future prices. If this model does not give a good enough result, we may have to extend the model to include $X_{t-1} - X_{t-2}$ or a_{t-1} and so forth.

The above example illustrates very descriptively the way a time series is analyzed. Our main concern in the following chapters is "how to estimate good parameters from finite number of data". "Goodness" means better fitting to data with less number of parameters. $\{Z_t\}$ is assumed to have no periodic or seasonal characters in this document. (The stock market usually shows a seasonal change, for example.)

From now on a time series is denoted by $\{Z_t\}$ if its average is not zero and $\{\tilde{Z}_t\}$ if it is adjusted so that its average becomes zero. The residual random term is denoted by $\{a_t\}$. It is assumed that $\{a_t\}$ is not correlated and is usually assumed to be Gaussian $E\{a_t\} = 0$, $\mathrm{Var}\{a_t\} = \sigma_a^2$.

There are several types of models conceivable - autoregressive models, moving average models and mixed models, each of which will be introduced in the following sections of this chapter, [1].

1.1 Autoregressive Models (AR Model)

In this model, the time series $\{\tilde{Z}_t\}$ is described by the following formula:

$$\tilde{Z}_t = \sum_{j=1}^{\infty} \phi \tilde{Z}_{t-j} + a_t$$

That is, $\tilde{Z}_t$ is a linear combination of previous values of the process itself and current random pulse. If B is defined as a backward shift operator (i.e. $B\tilde{Z}_t = \tilde{Z}_{t-1}$), (1) reduces to

$$(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3- \ldots.) \tilde{Z}_t = a_t \text{ or in short } \Phi(B)\tilde{Z}_t = a_t. \qquad (1)$$

where $E\{a_t\} = 0$. $E\{a_t^2\} = \sigma_a^2$.

When we build an AR model, $\Phi(B)$ is a polynomial of finite order, say p. In particular, the AR process of first order and of second order

$$\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + a_t$$

$$\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + \phi_2 Z_{t-2} + a_t$$

are important in practice.

1.2 Moving Average Models (MA Model)

This model is described by the following formula

$$\tilde{Z}_t = a_t + \sum_{j=1}^{\infty} \theta_j a_{t-j}$$

$$= (1+\theta_1 B +\theta_2 B^2 + \ldots) a_t$$

$$= \theta(B) a_t \qquad (2)$$

This is called "moving average" because Z_t is a weighted sum of another process a_t. Hence, we can regard $\{\tilde{Z}_t\}$ as the output of a linear filter of which $\{a_t\}$ is the input.

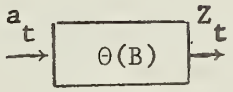


Figure 1.1

$\theta(B)$ is a polynomial of finite order q when we build a moving average model. In particular, the process of first order and of second order

$$\tilde{Z}_t = a_t - \phi_1 a_{t-1}$$

$$\tilde{Z}_t = a_t - \phi_1 a_{t-1} - \phi_2 a_{t-2}$$

are important in practice.

1.3 Mixed Models

As one can see, the moving average models and autoregressive models are in close relation - one can convert from one model to another. In other words (1) can be written as

$$Z_t = \Phi^{-1}(B)\, a_t \quad \text{and same is true with (2).}$$

An autoregressive model of the finite order, however, becomes of infinite order in the Moving Average Model and vice versa. As an extention of two methods, it is natural to consider models as follows:

$$\Phi(B)\, Z_t = \theta(B)\, a_t \tag{3}$$

This model is of infinite order in both MA model and AR model and it is to be hoped that in some cases we can describe a model with less number of parameters than either MA or AR models. Mixed process with $\Phi(B)$ of order p and $\theta(B)$ of order q is denoted by ARMA (p,q). In practice, ARMA (1,1)

$$\tilde{Z}_t - \phi_1 \tilde{Z}_{t-1} = a_t - \phi_1 a_{t-1}$$

is important.

1.4 Stationarity and Invertibility

Let us consider a first order autoregressive model

$$Z_t = a_t + \alpha Z_{t-1}.$$

If $|\alpha| > 1$, Z_t starting from any value will either grows bigger explosively or oscillates with increasing amplitude as time goes on. That is, a_t does not have enough power to retain Z_t around its average and the process becomes almost deterministic.

A model of this kind is called "non-stationary". This case will not be discussed in this document.

On the other hand, if $|\alpha| < 1$ the effect of previous values of Z die out with time, and a_t can always play an important role in the process.

In general, an autoregressive process

$$\Phi(B)\, \tilde{Z}_t = a_t$$

is called stationary if the infinite series $\Phi^{-1}(B)$ is convergent on and within the unit circle, or equivalently, if all the zeroes of $\Phi(B)$ are located outside the unit circle. The explosive processes can be stated as the cases in which some zeroes of $\Phi(B)$ are located within the unit circle.

Sometimes $\Phi(B)$ has several zeroes <u>on</u> the unit circle but none are within it. It turns out that the resulting models are of great value in representing some nonstationary series which may nevertheless exhibit <u>homogeneous</u> behaviour. In particular, although the general level about which fluctuations are occurring may be different at different times, the broad behaviour of the series, when differences in level are allowed for may be similar. Many series actually encountered in industry or business exhibit somewhat nonstationary behaviour which can be handled with this type of model.

The polynomial of autoregressive part, in such a case, can be written as

$$\Phi(B) = \phi(B)\, (1-B)^d$$

where $\phi(B)$ is a strictly stationary operator.

Thus a general model is

$$\phi(B)\, (1-B)^d\, \tilde{Z}_t = \Theta(B)\, a_t \qquad (4)$$

or $\phi(B)\, W_t = \Theta(B)\, a_t$

where $W_t = (1-B)^d\, \tilde{Z}_t$.

Homogeneous nonstationary behaviour can therefore be represented by a model which calls for the d^{th} difference of the process. In practice, d = 0, 1 or 2. This model is denoted by ARIMA (p,d,q) if $\phi(B)$ is of order p and $\theta(B)$ of order q. ARIMA is an abbreviation of Auto regressive integrated moving average model. The process is

$$W_t = \phi_1 W_{t-1} + \ldots + \phi_p W_{t-p} + a_t - \theta_1 a_{t-1} \ldots\ldots \theta_q a_{t-q}$$

with $W_t = \nabla^d Z_t$.

The general autoregressive-integrated-moving average process may be generated from white noise a_t by means of three filtering operations as shown below,

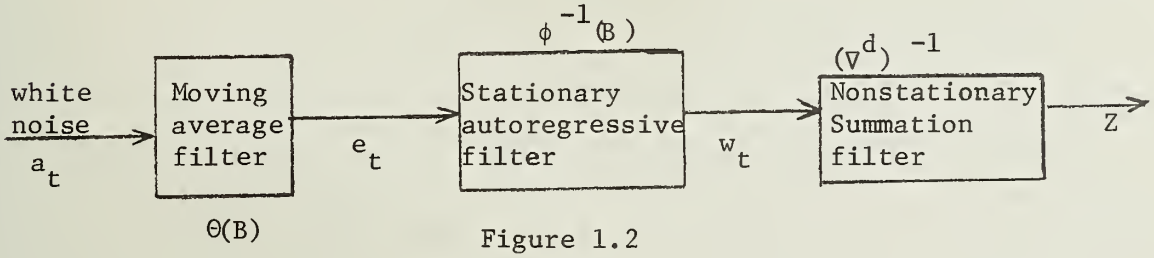


Figure 1.2

There is a corresponding property to stationarity in the moving average method. That is called "invertibility". Let us consider a first order moving average model to illustrate the basic idea of invertibility.

$$Z_t = (1-\theta B)\, a_t \qquad (5)$$

Expressing a_t's in terms Z_t's yields

$$a_t = (1+\theta B + \theta^2 B^2 + \ldots) Z$$

$$\text{or} \quad Z_t = a_t - \theta Z_{t-1} - \theta^2 Z_{t-2} \ldots\ldots\ldots \qquad (6)$$

The process (5) itself is definitely stationary for any θ. However, if $|\theta| > 1$, the weights in the corresponding autoregressive process (6) grow bigger as j in Z_{t-j} increases. We can avoid this ridiculous situation by putting $|\theta| < 1$. In general, a MA process $Z_t = \theta(B)$ is called "invertible" if $\theta^{-1}(B)$ is convergent for $|B| < 1$, or if all the zeroes of $\theta(B)$ are outside of the unit circle.

As we shall discuss later, a convergent expansion for a_t is possible when the above conditions are not satisfied, but only in terms of present and future values of the process. The invertibility is needed if we are interested in

associating present events with past happenings.

In the following discussion, both conditions for stationarity and invertibility are to be imposed on parameters of models.

Chapter 2 Linear Stationary Models

The first section is devoted to the autocorrelation function ρ_k, which is a very powerful tool in analyzing the time series. Linear stationary and non-stationary models which were introduced in Chapter 1 are discussed in more detail in the following sections. The formulae which are derived in this chapter are to be used in the subsequent chapters in estimating the parameters.

2.1 Autocorrelation Function

When a time series is stationary, that is, in a state of statistical equilibrium, we can define the mean value, variance and autocorrelation. Since the process is stationary, the distribution $p(Z_t)$ is the same for all times t and can be written $p(Z)$. Hence the process has a constant mean

$$\mu = E[Z_t] = \int_{-\infty}^{\infty} Z p(Z) dZ$$

and a constant variance

$$\sigma_z^2 = E[(Z_t - \mu)^2] = \int_{-\infty}^{\infty} (Z-\mu)^2 p(Z)\, dZ$$

If we have N observations $\{Z_1, \ldots . Z_n\}$ we can estimate μ by the mean of the time series

$$\bar{Z} = \frac{1}{N} \sum_{t=1}^{N} Z_t \qquad (1)$$

and the variance σ_2^2 by the simple variance of the time series

$$\hat{\sigma}_z^2 = \frac{1}{N} \sum_{t=1}^{N} (Z_t - \bar{Z})^2. \qquad (2)$$

The condition of stationarity also guarantees that the joint distribution $p(Z_{t_1}, Z_{t_2})$ is the same for all t_1 and t_2 if they are constant apart, i.e.

if a time series satisfies the condition $E\{|Z_t|^2\} < \infty$ for all t and $E\{Z_{t+s} Z_s\} = R(t)$ does not depend on s, it is called stationary in wide sense.

Then, we can define the autocovariance γ_k at lag k by,

$$\gamma_k = \text{cov}\,[Z_t, Z_{t+k}] = E\,[(Z_t-\mu)(Z_{t+k}-\mu)].$$

If Z_t is an ARIMA process, the autocovariance can be derived from the generating function. This is to be discussed in the next section.

The sample autocovariance C_k is the estimate to γ_k:

$$C_k = \frac{1}{N} \sum_{I=1}^{N-k} (Z_I - \bar{Z})\,(Z_{I+k} - \bar{Z}). \qquad (3)$$

The autocorrelation ρ_k at lag k is

$$\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{E\,[(Z_t-\mu)\,(Z_{t+k}-\mu)]}{\sigma_z^{\,2}} \qquad (4)$$

where $\rho_0 = 1$. The corresponding sample autocorrelation is $\gamma_k = C_k/C_o$.

Covariance matrix Γn associated with a stationary process for observations made at n successive times is

$$\Gamma n = \begin{bmatrix} \gamma_o & \gamma_1 & \gamma_2 \cdots\cdots \gamma_{n-1} \\ \gamma_1 & \gamma_0 & \gamma_{n-2} \\ \vdots & \ddots & \gamma_1 \\ \gamma_{n-1} & \gamma_{n-2} & \gamma_1 \;\; \gamma_0 \end{bmatrix} = \sigma_z^{\,2} \begin{bmatrix} 1 & \rho_1 & \rho_2 \cdots\cdots \rho_{n-1} \\ \rho_1 & 1 & \\ \rho_2 & & 1 \\ \vdots & & \ddots \\ \rho_{n-1} & & 1 \end{bmatrix}. \qquad (5)$$

The matrix is non-negative definite, but takes 0 for trivial cases.

(Example) Calculate $\bar{Z}$, C_1, γ_1, of the data given below.

$$Z = \{47,\ 64,\ 23,\ 71,\ 38,\ 64,\ 55,\ 41,\ 59,\ 48\}$$

$$\bar{Z} = \frac{1}{10}\ (47 + 64 + \ldots.) = 51$$

$$\tilde{Z} = (-4,\ 13,\ -28,\ 20,\ -13,\ 13,\ 4,\ -10,\ 8,\ -3).$$

$$C_0 = \frac{1}{10}\ (4^2 + 13^2 + \ldots + 3^2) = 189.6$$

$$C_1 = \frac{1}{10}\ ((-4) \times 13 + 13 \times (-28) + \ldots. + 8 \times (-3)) = -1497$$

$$\gamma_1 = \frac{C_1}{C_0} = -0.79$$

Estimated values of the autocorrelations are also stochastic variables and their standard errors are very important when we want to know whether or not the theoretical autocorrelations are truncated at some lag q and larger. Naturally the error becomes larger if size of the data is not sufficiently large. Bartlett [2] gave a formula for the variance of estimated autocorrelation when it is known that only $\rho_1, \rho_2 \ldots . \rho_q$ are non zero. That is

$$\text{Var}\,[\gamma_k] \simeq \frac{1}{N}\{1 + 2\sum_{u=1}^{q} \rho_u^2\} \qquad \text{if } k \geq q \tag{6}$$

Therefore, we can substitute $\gamma_1 \ldots . \gamma_q$ for $\rho_1 \ldots . \rho_q$, respectively to see whether the assumption that $\rho_k = 0$ $(k > q)$ is valid or not. This method will appear in the identification procedure of parameter estimation.

2.2 Autocovariance Generating Functions of ARIMA Processes

There is a very convenient formula to give any order of autocovariance function. Suppose an ARIMA process is given as

$$\Phi(B)\, Z_t = \Theta(B)\, a_t .$$

Operation $\Phi^{-1}(B)$ from the left yields

$$Z_t = \Phi^{-1}(B)\Theta(B)a_t = \Psi(B)\, a_t$$

$$= \sum_{j=0}^{\infty} \Psi_j B^j a_t = \sum_{j=0}^{\infty} \Psi_j a_{t-j}$$

$$\gamma_k = E[Z_t Z_{t+k}] = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \Psi_i \Psi_j \underbrace{E[a_{t-i}\, a_{t+k-j}]}_{\sigma_a^2 \delta_{i,\, j-k}}$$

$$= \sigma_a^2 \sum_{j=0}^{\infty} \Psi_j \Psi_{j+k}. \tag{7}$$

Especially

$$\gamma_0 = \sigma_z^2 = \sigma_a^2 = \sum_{j=0}^{\infty} \Psi_j^2 . \tag{8}$$

The autocovariance generating function is defined as

$$\gamma(B) = \sum_{k=-\infty}^{\infty} \gamma_k B^k = \sigma_a^2 \sum_{k=-\infty}^{\infty} \sum_{j=0}^{\infty} \psi_j \psi_{j+k} B^k$$

Since $\psi_k = 0$ for all $k < 0$,

$$\gamma(B) = \sigma_a^2 \sum_{j=0}^{\infty} \sum_{h=0}^{\infty} \psi_j \psi_h B^{h-j} \qquad (h = j+k)$$

$$= \sigma_a^2 \sum_{j=0}^{\infty} \psi_j B^{-j} \sum_{h=0}^{\infty} \psi_h B^h$$

$$= \sigma_a^2 \, \Phi^{-1}(F) \, \Theta(F) \cdot \Phi^{-1}(B) \Theta(B) \qquad (9)$$

(Example) MA(1) Process

$$\tilde{Z}_t = a_t - \theta a_{t-1} = (1-\Theta B) \, a_t$$

$$\gamma(B) = (1-\Theta B^{-1})(1-\Theta B)$$

$$= 1 + \Theta^2 - \Theta B^{-1} - \Theta B$$

$$\therefore \begin{cases} \gamma_0 = 1+\Theta^2 \\ \gamma_1 = -\Theta \\ \gamma_k = 0 \qquad (k \geq 2). \end{cases}$$

2.3 More About Autoregressive Models (Stationary)

An autoregressive model of order p, AR (p), is denoted by

$$\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + \phi_2 \tilde{Z}_{t-2} + \ldots\ldots + \phi_p \tilde{Z}_{t-p} + a_t \qquad (10)$$

or in short $\Phi(B) \tilde{Z}_t = a_t$

where $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots\ldots$

2.3.1 Autocorrelations of Autoregressive Models (Yule-Walker equation)

Multiplying (10) by $\tilde{Z}_{t-k}$ and taking the expectation at each term yields

$$\gamma_k = \phi_1 \gamma_{k-1} + \phi_2 \gamma_{k-2} + \ldots + \phi_p \gamma_{k-p} \qquad k > 0 \qquad (11)$$

because $E[Z_{t-k}\, a_t] = 0$ for $k > 0$. Note that the expectation vanishes when $k < 0$.

Hence

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \ldots\ldots \phi_p \rho_{k-p} \qquad (12)$$

As can be seen from (12) the autocorrelations for AR processes is infinite in extent. If we take the first p equations in (12) and write them in matrix notation, we get:

$$\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{bmatrix} = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \cdots & \rho_{p-1} \\ \rho_1 & 1 & \rho_2 & \cdots & \rho_{p-2} \\ \vdots & \rho_1 & 1 & & \\ & & & & \\ \rho_{p-1} & & & & \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} \qquad (13)$$

(13) is usually called Yule-Walker equation and is very useful in the preliminary estimation of ϕ's from sample autocorrelations.

The variance σ_z^2 is obtained from (10) by taking its expectation after multiplying it by Z_t. Since $\tilde{Z}_t$ is related to a_t, $E\{Z_t\, a_t\} = \sigma_a^2$.

Hence
$$\sigma_z^2 = \frac{\sigma_a^2}{1 - \phi_1 \rho_1 - \phi_2 \rho_2 - \ldots\ldots - \phi_p \rho_p}$$

2.3.2 Examples

(1) $\tilde{Z}_t = \phi \tilde{Z}_{t-1} + a_t$ (AR(1) process)

This process is called Markov process.

$$\rho_1 = \phi \qquad (14)$$

$$\rho_k = \phi^k \quad (k > 1) \qquad (15)$$

$$\sigma_z^2 = \frac{\sigma_a^2}{1 - \rho_1 \phi} = \frac{\sigma_a^2}{1 - \phi^2} \qquad (16)$$

$|\phi| < 1$ (stationarity condition)

(2) $\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + \phi_2 \tilde{Z}_{t-2} + a_t$ (AR (2) Process)

The stationarity condition yields

$\phi_1 + \phi_2 < 1$

$\phi_2 - \phi_1 < 1$

$-1 < \phi_2 < 1$

$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2}$

$\rho_o = 1$

$\rho_1 = \frac{\phi_1}{1-\phi_2}$

$\rho_2 = \phi_2$

ϕ_1 and ϕ_2 are estimated from Yule-Walker equations:

$$\phi_1 = \frac{\rho_1 (1-\rho_2)}{1 - \rho_1^2}$$

$$\phi_2 = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2} \quad .$$

2.4 The Partial Autocorrelation Function

There is another powerful function which helps us check whether a process is autoregressive or not. Suppose a time series is described by AR (p) model. Then the autocorrelation function ρ_k, although infinitely extended, is a linear combination of p non zero functions of autocorrelations. For example, in AR (1) process $\tilde{Z}_t = \phi \tilde{Z}_{t-1} + a_t$ (See 2.3.2) $\rho_1 = \phi$ is the only independent function. So let us investigate how we can exploit this fact to identify the autoregressive models.

Let us assume that the process is of the order ℓ so that

$$\tilde{Z}_t = \phi_{\ell 1} \tilde{Z}_{t-1} + \phi_{\ell 2} \tilde{Z}_{t-2} + \ldots\ldots + \phi_{\ell\ell} \tilde{Z}_{t-\ell} + a_t \qquad (17)$$

From (12) we obtain

$$\rho_j = \phi_{\ell 1} \rho_{j-1} + \phi_{\ell 2} \rho_{j-2} + \ldots\ldots + \phi_{\ell\ell} \rho_{j-\ell} \qquad (18)$$

$$j = 1, 2 \ldots \ell$$

where we can solve $\phi_{\ell,j}$'s in terms ρ_j's by the Yule-Walker equations. We solve these equations for $\ell = 1,2,3,....$ successively to get a sequence (ϕ_{11}, ϕ_{22}, ϕ_{33},....). Since $\phi_{\ell\ell}$ is the coefficient of the last term in (17), it has a cut off after lag p if the original process is of degree p. Quenouille showed [10] that the variance of the estimated ϕ_{kk} from n data if $k > p$ is given by

$$\text{var}\,[\phi_{kk}] \simeq \frac{1}{n} \qquad \text{for } k \geq p + 1.$$

Hence we can know the degree of autoregressive models by plotting ϕ_{kk} against k and by checking when it starts to be submerged within the region defined by $\sqrt{\text{var}\,[\phi_{kk}]} = n^{-\frac{1}{2}}$. An example is given in Fig. 2.1.

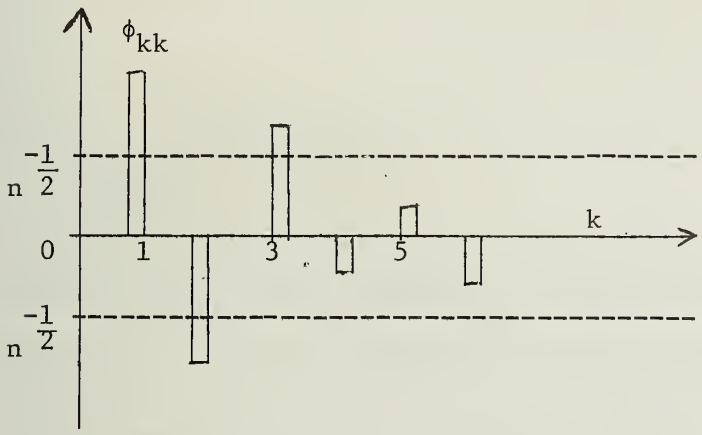


Figure 2.1
An example of typical behavior of the partial autocorrelations.

There is a very efficient algorithm to compute the partial autocorrelatio function without solving Yule Walker equations at each step. This is a recursive method in which the estimate for AR(p) can be computed from those of AR (p-1) and autocorrelations. The general recursive formulae are

$$\hat{\phi}_{p+1,\,j} = \hat{\phi}_{pj} - \hat{\phi}_{p+1,\,p+1}\,\hat{\phi}_{p,p-j+1} \qquad j = 1,2,3,...p \qquad (18)$$

$$\hat{\phi}_{p+1,p+1} = \frac{r_{p+1} - \sum_{i=1}^{p} \hat{\phi}_{pj}\, r_{p+1-j}}{1 - \sum_{i=1}^{p} \hat{\phi}_{pj}\, r_j} \qquad (19)$$

These formulae are actually used in the program USID.

2.5 More About Moving Average Models

A moving average model MA(q) is denoted by

$$\begin{aligned}\tilde{Z}_t &= a_t - \theta_1 a_{t-1} - \theta_2 \; t-2 \cdots\cdots \theta_q a_{t-q} \\ &= (1- \theta_1 B \cdots\cdots - \theta_q B^q) a_t \\ &= \theta(B) a_t \end{aligned} \tag{20}$$

2.5.1 Autocorrelation function of Moving Average Models

The autocovariance function of a MA(q) process is

$$\gamma_k = E[(a_t - \theta_1 a_{t-1} \cdots\cdots \theta_q a_{t-q})(a_{t-k} - \theta_1 a_{t-k-1} \cdots - \theta_q a_{t-k-q})] \tag{21}$$

Since $E[a_{t-k} a_{t-\ell}] = \sigma_a^2 \delta_{k\ell}$, (21) yields

Hence $$\gamma_o = (1+\theta_1^2 + \theta_2^2 + \cdots\cdots + \theta_q^2) \sigma_a^2$$

and $$\gamma_k = (\theta_k + \theta_1 \theta_{k+1} + \theta_2 \theta_{k+2} + \cdots\cdots + \theta_{q-k} \theta_q) \sigma_a^2 \quad (k=1,2,\ldots.q) \tag{22}$$

$$\gamma_k = 0 \qquad (k > q.)$$

We see that the autocorrelation function of a MA(q) process has a cut off at lag q. The difficulty with moving average models is that the equations (22) are not linear if we are to solve them with regard to θ's. Therefore, they must be solved by some iterative methods. The algorithm will be found in 3.3 and the actual program in Appendix.

2.5.2 Examples

(1) $Z_t = a_t - \theta_1 a_{t-1}$ (MA (1) process)

invertibility condition $-1 < \theta < 1$

$$\gamma_o = (1+ \theta_1^2) \sigma_a^2$$

$$\rho_1 = \frac{-\theta_1}{1+\theta_1^2}$$

$$\rho_k = 0 \qquad k > 2$$

$$\theta_1 = (-1 \pm \sqrt{1-4\rho_1^2})/2\rho_1$$

$$\phi_{kk} = - \theta_1^k \left\{ \frac{1 - \theta_1^2}{1 - \theta_1^{2(k-1)}} \right\}$$

(2) $\tilde{Z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}$ (MA (2) process)

Invertibility $\theta_2 + \theta_1 < 1$

$\theta_2 - \theta_1 < 1$

$-1 < \theta_2 < 1$

$$\gamma_o = (1+\theta_1^2 + \theta_2^2)\,\sigma_a^2$$

$$\rho_1 = \frac{-\theta_1 (1- \theta_2)}{1+ \theta_1^2 + \theta_2^2}$$

$$\rho_2 = \frac{- \theta_2}{1 + \theta_1^2 + \theta_2^2}$$

$$\rho_k = 0 \qquad k \geq 3.$$

2.6 Duality

One will notice the close relation between AR models and MA models. As has been discussed in Chapter 1, an AR model of finite order becomes a MA model of infinite order and vice versa. Therefore, the autocorrelations of a MA (12) process have a cut off at lag k, whereas its partial autocorrelations are extended to infinity because the process is equivalent to an AR process of infinite order. Note also that the dual of the stationary condition in AR models is the invertibility condition in MA models. The duality will become more obvious in Mixed Models.

2.7 More About Mixed Models

We have noted in Chapter 1 that in some cases mixed models require less number of parameters than MA models or AR models because the mixed models have the characteristics of both.

We employ the mixed models as follows:

$$\tilde{Z}_t = \theta_1 \tilde{Z}_{t-1} + \ldots\ldots + \theta_p \tilde{Z}_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} \ldots \theta_q a_{t-q} \qquad (23)$$

or in short

$$\Phi(B)\, \tilde{Z}_t = \Theta(B)\, a_t \qquad (24)$$

This process is referred to as ARMA (p,q).

(24) defines a stationary process, provided that all zeroes of $\Phi(B)$ are outside the unit circle, and defines an invertible process if all zeroes of $\Theta(B)$ are outside the unit circle. We assume that these conditions are satisfied. Remember that $\Phi(B)$ has some zeroes on the unit circle when we refer to ARIMA processes.

2.7.1 Autocorrelation function of mixed models

On multiplying (2.7.1) by $\tilde{Z}_{t-k}$ and taking the expectation yields

$$\gamma_k = \phi_1 \gamma_{k-1} + ... + \phi_p \gamma_{k-p} + \gamma_{za}(k) - \Theta_1 \gamma_{za}(k-1) ... - \Theta_q \gamma_{za}(k-q)$$

where $\gamma_{za}(k) = E[Z_{t-k} a_t]$ is the cross covariance function between z and a.

Obviously $\gamma_{za}(k) = 0$ if $k > 0$.

Therefore

$$\gamma_k = \Theta_1 \gamma_{k-1} + \Theta_2 \gamma_{k-2} + ... + \Theta_p \gamma_{k-p} \qquad (k > q+1)$$

Hence

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + ... + \phi_p \rho_{k-p}$$

Or $$\phi(B) \rho_k = 0 \qquad (k > q+1)$$

Thus, autoregressive parameters are separated from moving average parameters if we take lag $k > q+1$. We can derive a modified MA (q) process

$$\tilde{Z}'_t = \tilde{Z}_t - \phi_1 \tilde{Z}_{t-1} - \phi_2 \tilde{Z}_{t-2} .... - \phi_q Z_{t-q}$$

if we know all the autoregressive parameters. This fact is used in the preliminary estimation of parameters. (Chapter 3).

2.7.2 (Example)

ARMA (1,1) process

Practically, this process is very important.

$$\tilde{Z}_t - \phi_1 \tilde{Z}_{t-1} = a_t - \Theta_1 a_{t-1}.$$

or $$(1 - \phi_1 B) \tilde{Z}_t = (1 - \Theta_1 B) a_t .$$

Autocorrelations are

$$\gamma_o = \phi_1 \gamma_1 + \sigma_a^2 - \theta_1 \gamma_{za}(-1)$$

$$\gamma_1 = \phi_1 \gamma_o - \theta_1 \sigma_a^2$$

$$\gamma_k = \phi_1 \gamma_{k-1} \qquad k \geq 2.$$

It is easy to find $\gamma_{za}(-1)$.

$$\gamma_{za}(-1) = (\phi_1 - \theta_1)\sigma_a^2$$

Therefore

$$\gamma_o = \frac{1 + \theta_1^2 - 2\phi_1\theta_1}{1 - \phi_1^2}\sigma_a^2$$

$$\gamma_1 = \frac{(1-\phi_1\theta_1)(\phi_1 - \theta_1)}{1 - \phi_1^2}\sigma_a^2$$

Or

$$\rho_1 = \frac{(1-\phi_1\theta_1)(\phi_1 - \theta_1)}{1+\theta_1^2 - 2\phi_1\theta_1}$$

$$\rho_2 = \phi_1\rho_1$$

from which ϕ_1, ϕ_2, and σ_a^2 can be calculated.

Finally, it is worthwhile to summarize the typical behavior of autocorrelations and partial autocorrelations of three models. If the process is AR then a.c. has exponential decay or damped oscillations whereas p.a.c. has a sharp cut off. If the process is MA, then the situation is reversed. In MIX processes both a.c. and p.a.c. decay exponentially or have damped oscillation (there are four combinations depending on the value of parameters). (See Section 3.1 for a summary.)

Appendix A1. Backward Representations of Time Series

Suppose that W_t is represented by the linear model

$$\phi(B)\, W_t = \Theta(B)\, a_t ,$$

where the zeroes of $\phi(B)$ and $\Theta(B)$ lie outside the unit circle. It is very interesting to note that the autocovariance generating function $\gamma(B)$ (§2.2), given by,

$$\gamma(B) = \Phi(F)^{-1}\ \Theta(F)\ \Phi^{-1}(B)\ \Theta(B)$$

is invariant under the exchanges $B \to F$ and $F \to B$. Thus the stochastic process

$$\phi(F)\, W_t = \Theta(F) e_t \qquad (1)$$

(even though its meaning is not clear as yet) has the identical covariance structure as the original stochastic process

$$\phi(B)\, W_t = \Theta(B)\, a_t . \qquad (2)$$

e_t in (1) is a sequence of independently distributed random variables having mean zeros and variances $\sigma_e^2 = \sigma_a^2$.

The stochastic process (1), means that W_t can be expressed entirely in terms of future W's and e's and is a stationary, invertible representation. We refer to it as the backward form of the process. We will use it in the estimation of the parameters.

Appendix A2. Spectral Properties

The autocorrelation function is not the only tool to analyze a time series with; sometimes the power spectrum is very useful. Actually in the fields such as oceanography or communication engineering, the power spectrum is used rather than the autocorrelation function.

In this document, however, the power spectrum appears only in Chapter 6 to check the periodicity of a time series, therefore a very brief review is presented here. The reader should refer to [3] for more details on this subject.

1) Periodgram

In the analysis of the spectral properties we regard a given time series as composed of sine and cosine waves of different frequencies. First we define periodogram of a time series. That is, if the number of the observation is N = 2q+1. (odd)

$$Z_t = a_o + \sum_{i=1}^{q} (a_i \cos 2\pi f_i t + b_i \sin 2\pi f_i t) + e_t$$

where $f_i = \frac{i}{N}$ is the ith harmonic of the fundamental frequency 1/N. Fourier coefficients a_o, a_i, b_i, are estimated as

$$a_o = \overline{Z_t}$$

$$a_i = \frac{2}{N} \sum_{t=1}^{N} Z_t \cos 2\pi f_i t, \quad b_i = \frac{2}{N} \sum_{t=1}^{N} Z_t \sin f_i t \qquad t=1,2,3,\ldots N.$$

The periodogram then consists of $q = \frac{N-1}{2}$ values

$$I_i = \frac{N}{2} (a_i^2 + b_i^2) \qquad i = 1,2,\ldots,q$$

When N is even, we set N = 2q and obtain

$$a_o = \overline{Z}$$

$$a_i = \frac{2}{N} \sum_{t=1}^{N} Z_t \cos 2\pi f_i t \qquad i=1,2,\ldots q-1$$

$$a_q = \frac{1}{N} \sum_{t=1}^{N} (-1)^t Z_t$$

$$b_i = \frac{2}{N} \sum_{t=1}^{N} Z_t \sin 2\pi f_i t$$

$$b_q = 0$$

$$I_i = \frac{N}{2}(a_i^2 + b_i^2) \qquad i=1,2,\ldots q-1$$
$$I_q = I(.5) = Na_q^2$$

Note that the highest frequency is .5 per time interval.

I_i indicates the relative intensity of the ith frequency component. If Z_t contains $a_i \cos 2\pi f_i t$ or $b_i \sin 2\pi f_i t$ or both, the periodogram shows a peak at i. On the other hand, if $Z_t = \alpha_o + e_t$, the periodogram is constant. This is why we can discuss some periodic features of a time series by using its spectral properties.

2) Sample spectrum and power spectrum

I_i assumes a discrete argument. We define I(f) to have a continuous argument f and to take values I_i at $f_i = \frac{i}{N}$. This new function is called the sample spectrum. There is a very important relation between I(f) and the autocovariance function C_k of Z_t.

Theorem. The sample spectrum is the Fourier cosine transform of the auto-covariance function:

$$I(f) = 2\{C_0 + 2\sum_{K=1}^{N-1} C_k \cos 2\pi f_k\} \qquad 0 < f < \frac{1}{2}$$

Proof:

$$I(f) = \frac{N}{2}(a_f^2 + b_f^2) = \frac{N}{2}(a_f + ib_f)(a_f - ib_f)$$
$$= \frac{N}{2} d_f^* d_f$$

where

$$d_f = a_f - ib_f = \frac{2}{N}\sum_{t=1}^{N}(Z_t \cos 2\pi ft - iZ_t \sin 2\pi ft)$$
$$= \frac{2}{N}\sum_{t=1}^{N}(Z_t - \bar{Z})\, e^{-2\pi fit}$$

$$\therefore\ I(f) = \frac{2}{N}\sum_{t=1}^{N}\sum_{t'=1}^{N}(Z_t - \bar{Z})(Z_{t'} - \bar{Z})\, e^{-2\pi if(t-t')}$$

$$= \frac{2}{N}\sum_{K=(n-1)}^{N-1}\sum_{j=1}^{N-k}(Z_j - \bar{Z})(Z_{j+k} - \bar{Z})\, e^{-2\pi ifk}$$

$$= 2\sum_{k=-(N-1)}^{N-1} C_k\, e^{-2\pi ifk}$$

$$= 2\{C_o + 2\sum_{k=1}^{N-1} C_k \cos 2\pi fk\}.$$

I[f] is an estimate which was obtained from N samples $\{Z_1, Z_2, \ldots . Z_n\}$. It is a stochastic variable. We take the expectations value of I(f),

$$E\{I[f]\} = 2\{E[C_0] + 2 \sum_{k=1}^{N-1} E[C_k] \cos 2\pi fk\}$$

As N goes to infinity, $E[C_i]$ tends to the theoretical autocovariance function γ_i. Therefore, we can define a power spectrum p(f) as a limit of $E\{I[f]\}$:

$$p(f) = \lim_{n \to \infty} E\{I(f)\} = 2\{\gamma_o + 2\Sigma \gamma_k \cos 2\pi fk\} .$$

Appendix A3. Conditional Expectation

The conditional expectation of Z_t at time k, $E_k[Z_t]$ is the expectation given complete historical knowledge up to, but not beyond time k. For the simplicity in notation, let square brackets imply that the conditional expectation at time t is to be taken. For example,

$$[a_{t+\ell}] = \underset{t}{E}\,[a_{t+\ell}]$$

$$[Z_{t+\ell}] = \underset{t}{E}\,[Z_{t+\ell}].$$

Some properties are:

(1) $[Z_{t-j}] = \underset{t}{E}\,[Z_{t-j}] = Z_{t-j}$ $\qquad j = 0,1,2,....$

That is to say, standing at time t the expected values of Z's that have happened already are the values they have actually realized.

(2) $[Z_{t+j}]$ = prediction of Z_{t+j} at time t.

(3) $[a_{t-j}] = \underset{t}{E}\,[a_{t-j}] = a_{t-j}$

(4) $[a_{t+j}] = \underset{t}{E}\,[a_{t+j}] = 0.$

i.e., at time t, the expected values of the a's that have yet to happen are zero.

Chapter 3 Identification Of Models

Now we are prepared to tackle practical problems; given a time series, we can estimate the parameters. There are three stages for analyzing the time series. Firstly we must identify the type and the order of the process, we can obtain crude approximation to the parameters at this stage. Secondly we must improve the estimate of the parameters. And finally we must check whether or not the estimated model really fits the given time series. This chapter is concerned with the first step, i.e. identification of the model. Our principal tools are the autocorrelations and the partial autocorrelations.

It is inevitable that identification and estimation overlap. To identify a model, we must make some inference on parameters. But in this stage the estimate is inexact. Our aim here is to identify the model to a given time series in the following way:

1. to difference Z_t as many times as is needed to produce stationarity, reducing the process under study to the ARMA process

$$\phi(B)\, W_t = \theta_o + \theta(B)\, a_t$$

where $W_t = (1-B)^d Z_t$.

2. to identify the resulting ARMA process.

Programs are also given in appendices. They are explained in the text.

3.1 Use of Autocorrelations and Partial Autocorrelations in Identification Procedure

3.1.1. Identifying the degree of differencing.

We have seen in Chapter 2 that the autocorrelations die out quickly if none of roots $\phi(B) = 0$ are close to the unit circle. Therefore, if the plotted autocorrelations do not show quick decay, it indicates that further differencing is necessary. In practice the order of differencing is 0,1 or 2 and it is usually sufficient to inspect the first 20 or so estimated autocorrelations.

3.1.2. Identification of resultant stationary ARMA process

Having tentatively decided the order of differencing, we next study the general appearance of the estimated autocorrelations and partial autocorrelations of the appropriately differenced series. The characteristic behavior of these functions summarized in Chapter 2 is very helpful to

identify the type of the model. Both autocorrelations and partial autocorrelations should be plotted for first 20 lags or so. This can be nicely done by line printers or CRT displays. (See the appendix).

Some important characteristics of the models are summarized in Table 3.1.

TABLE 3.1

order	(1, d, 0)	(o, d, 1)
ρ_k	decays exponentially	only ρ_1 non-zero
ϕ_{kk}	only ϕ_{11} non-zero	exponential dominates decay
estimates	$\phi_1 = \rho_1$	$\rho_1 = \frac{-\theta_1}{1+\theta_1^2}$
admissible region (stationarity, invertibility)	$-1 < \phi_1 < 1$	$-1 < \theta_1 < 1$
order	(2, d, 0)	(0, d, 2)
ρ_k	mixture of exponential or damped sine wave	only ρ_1 and ρ_2 non-zero
ϕ_{kk}	only ϕ_{11} and ϕ_{22} non-zero	dominated by mixture of exponential or damped sine wave
estimates	$\phi_1 = \frac{\rho_1 (1-\rho_1)}{1-\rho_1^2}$	$1 = \frac{-\theta_1 (1-\theta_2)}{1+\theta_1^2+\theta_2^2}$
	$\phi_2 = \frac{\rho_2 - \rho_1^2}{1-\rho_1^2}$	$2 = \frac{-\theta 2}{1+\theta_1^2+\theta_2^2}$
admissible region	$-1 < \phi_2 < 1$	$-1 < \theta_2 < 1$
	$\phi_1 + \phi_2 < 1$	$\theta_2 + \theta_1 < 1$
	$\phi_2 - \phi_1 < 1$	$\theta_2 - \theta_1 < 1$

	(1, d, 1)
ρ_k	decays exponentially from first lag
ϕ_{kk}	dominated by exponential decay from first lag
estimates	$\rho_1 = \dfrac{(1-\theta_1\phi_1)(\phi_1 - \theta_1)}{1+\theta_1^2 - 2\phi_1\theta_1}$ $\rho_2 = \rho_1\phi_1$
admissable region	$-1 < \phi_1 < 1, \qquad -1 < \theta_1 < 1$

3.1.3. Standard error of the autocorrelations and the partial autocorrelations

Usually the estimated autocorrelations and partial autocorrelations fairly fluctuate around the axis and take moderately large values after they once decay. Therefore we must have a broad view in identifying models. Sometimes we may have multiple models until estimation stage or even until diagnosis stage. For large lags there are some formulae which give clues to the standard errors of those functions. For example, if the theoretical autocorrelations are 0 after lag q, the standard error of the estimated autocorrelations is given as

$$\sigma[r_k] \simeq \frac{1}{\sqrt{n}} \left(1+2\left(r_1^2 + r_2^2 + \cdots + r_q^2\right)\right)^{1/2} \qquad k > q \quad \text{(see (2.1))}.$$

As for the partial autocorrelation function, the estimated function has the standard error

$$\hat{\sigma}[\phi_{kk}] \simeq \frac{1}{\sqrt{n}} \qquad k > p \tag{2.4}$$

if the theoretical function is 0 for $k > p$.

Therefore we can draw 1σ or 2σ lines in the graph of those functions to indicate whether they are negligible or not.

3.2 Description of program USID

This program is for the identification of the models.

1) Input parameters

   Z: time series to be analyzed (saved)

   N: number of data

   CD: maximum lag of autocorrelation

   L: maximum lag of partial autocorrelations

   D: order of higher difference

2) Output

   MEAN: mean value of Z ($\bar{Z}$)

   VAR: variance of Z (C_0)

   ATCR: Autocorrelations

   $\sigma[\gamma_k]$: Standard error of autocorrelations

   $\sigma[\phi_{kk}]$: Standard error of partial autocorrelations

3) Subroutine

   CORRECOPRINT: print out the function

4) Function

   Read as data.

   Take the higher difference as specified (D)

   Calculate the mean value, variance, auto and partial autocorrelations.

   Calculate the standard error of them.

   Plot the result, via a line printer

3.3 Preliminary Estimates of Parameters

Now that we have an appropriately differenced time series $W_t = \nabla^d Z_t$, we can make initial estimates of parameters. Formulae for lower models are given in Table 3.1.

There are general formulae which work for any order and which are very suitable for computer programming. They will be shown in the subsequent sections.

3.3.1 AR Parameters

Autoregressive parameters are obtained from Yule-Walker equations:

$$\hat{\phi} = R_p^{-1} r_p \quad . \tag{2.13}$$

This is a system of linear equations and is solved by any standard method. (It is recommended to have a subprogram to solve a system of linear equations in one's programs.) It can be proved that the autoregressive parameters obtained from Yule-Walker equations for the estimated autocorrelations approximate the fully efficient maximum likelihood estimates. (Proof in [1].)

The initial estimate of the residual variance is

$$\hat{\sigma}_a^2 = Co\ (1 - \hat{\phi}_1\ r_1 - \hat{\phi}_2\ r_2 - \ldots\ldots - \hat{\phi}_p\ r_p).$$

3.3.2 MA Parameters

In general the equations for moving average parameters are non-linear and much harder to solve than AR parameters. More explicitly, they are

$$r_k = \frac{-\hat{\theta}_k + \hat{\theta}_1\ \hat{\theta}_{k+1} + \ldots\ \hat{\theta}_{q-k}\ \hat{\theta}_q}{1 + \hat{\theta}_1^2 + \hat{\theta}_2^2 + \ldots + \hat{\theta}_q^2} \qquad k = 1,2,\ldots.q$$

$$C_0 = \hat{\sigma}_a^2\ (1 + \theta_1^2 + \ldots + \theta_q^2)$$

This system of nonlinear equations is solved iteratively. The method discussed here is a Newton-Raphson's algorithm which converges quadratically. Suppose we have the estimated autocovariances, $C_0, C_1, C_2 \ldots C_q$.

Procedures:

1. $\tau_0 = \sqrt{C_0}, \quad \tau_1 = \ldots = \tau_q = 0$

2. Solve

$$\begin{bmatrix} \tau_0^{i+1} \\ \tau_1^{i+1} \\ \vdots \\ \tau_q^{i+1} \end{bmatrix} = \begin{bmatrix} \tau_0^{i} \\ \tau_1^{i} \\ \vdots \\ \tau_q^{i} \end{bmatrix} - T^{-1}\ f^i$$

where

$$T = \begin{bmatrix} \tau_0 & \tau_1 \cdots\cdots \tau_q \\ & \tau_0 \quad\quad \tau_{q-1} \\ & \ddots \quad \ddots \quad \vdots \\ & \ddots \quad \vdots \\ 0 & \end{bmatrix} + \begin{bmatrix} \tau_0 & \tau_1 \cdots\cdots \tau_q \\ \tau_1 & \tau_2 \cdots\cdots \tau_q \\ & \ddots \quad \tau_q \\ \tau_q & \ddots \quad\quad 0 \end{bmatrix}$$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_q \end{bmatrix} \qquad f_i = \sum_{j=0}^{q-j} \tau_i \, \tau_{i+j} - C_j$$

for τ^{i+1} iteratively until $|f_i|$'s become sufficiently small.

3. Finally $\begin{bmatrix} \hat{\theta}_1 \\ \hat{\theta}_2 \\ \vdots \\ \hat{\theta}_q \end{bmatrix} = \begin{bmatrix} -\tau_1/\tau_0 \\ -\tau_2/\tau_0 \\ \vdots \\ -\tau_q/\tau_0 \end{bmatrix}$

$$\hat{\sigma}_a^2 = \tau_0^2$$

This algorithm has been given by Wilson, [4].

3.3.3 Mixed Process

It is often found that $W_t = \nabla^d Z_t$ is most economically represented by a mixed ARMA process

$$\phi(B)\, W_t = \theta(B)\, A_t.$$

The procedure to estimate parameters of mixed ARMA process is a mixture of the two described in 3.3.1 and 3.3.2.

Suppose we are given the first p+q+1 autocovariances $C_0, C_1, \ldots, C_{p+q}$. Then we have three stages:

(1) The autoregressive parameters $\phi_1 \ldots\ldots \phi_p$ are estimated from $C_{q-p+1}, \ldots\ldots, C_{q+p}$. We discussed in 2.5.1 that if the lag is greater than q then only autoregressive parameters are related to autocovariance functions. Therefore, we can solve Yule-Walker equation for ϕ_i's.

(2) Using the estimates ϕ_i's obtained in (1), the first q+1 autocovariances C_j' (j = 0,1,...q) of the derived series $W_t' = W_t - \hat{\phi}_1 W_{t-1} \ldots - \hat{\phi}_p W_{t-p}$ are computed. We do not have to recalculate W_t''s to obtain C_j''s. The following formula yields directly the derived autocovariances.

$$C_j' = \sum_{i=0}^{P} \sum_{k=0}^{P} \phi_i \phi_k C_{|j+i-k|}, \qquad \hat{\phi}_o = -1, \ (j = 0,..,q).$$

(3) Finally the derived autocovariances $C_o' \ldots\ldots C_q'$ are used in an iterative calculation to compute initial estimates of the moving average parameters $\theta_1, \theta_2, .., \theta_q$ and the residual variance $\hat{\sigma}_a^2$.

3.4 Description of Program USPE

This program is for the initial estimates of parameters

1) Input parameters

W	time series
N	number of data
P	order of AR process
Q	order of MA process
CD	maximum lag for autocorrelation function
MEAN	mean value of w ($\overline{W}$)
VAR	variance of W (C_0)
ATCR	autocorrelation
EPSILØN	criterion for the convergence of Newton-Raphson's method

2) Output

PH1	AR parameters
THETA	MA parameters
SIGMA	Residual Variance

3) Subroutine

GAUSSLU Gaussian Elimination

4) Function

Compute AR parameters by Gauss elimination.

Compute derived autocovariances.

Compute T matrix and f.

Newton-Raphson's algorithm.

Compute MA parameters and residual variance.

Print out.

Chapter 4 Maximum Likelihood Estimates

The identification procedure has led to a tentative formulation for the model. We now need to obtain efficient estimates of the parameters. The method of the maximum likelihood function (essentially the method of least squares) is very suitable for this purpose.

The first section is an introduction to the maximum likelihood functions. Readers familiar with the subject can skip this section. Other sections of this chapter are devoted to the usage of the maximum likelihood functions to estimate parameters.

4.1 The likelihood functions *

The method of maximum likelihood is often a very suitable means of obtaining the required estimates. In many cases the errors of the estimates can be smaller than those which would arise from any other method of estimation. The concept of maximum likelihood estimators is introduced by some examples.

(Example 1 - discrete) Two urns are available, each containing black and white balls. Urn A has 3 black balls for every white ball, and urn B has 3 white balls for every black ball. One of the urns is chosen at random, and then three balls are picked up with replacement from the urn. The probability of getting r black balls is

$$P_3(r) = \binom{3}{r} p^r (1-p)^{3-r} \qquad (1)$$

where p = 3/4 for urn A
1/4 for urn B.

So far the problem is that of probability. Now, we want to estimate p on the basis of the observed example. (Note that the problem goes backward!) We can evaluate (1) for all possible values of r and for the two values of p. That is,

r	0	1	2	3
$P(r\|3/4)$	$1/64$	$9/64$	$27/64$	$27/64$
$P(r\|1/4)$	$27/64$	$27/64$	$9/64$	$1/64$

(*) This section is based on "Lectures on Elementary Statistics and Probability" by D. J. Hudson, CERN Report #63-29.

Suppose we in fact observe no black balls, i.e. r=0. We then argue that the observed data are 27 times more likely to have occurred if p = 1/4 than if p = 3/4. We may conclude that we must have chosen the urn B. We would say that the most likely estimate of p = 1/4 if r=0 or 1, and 3/4 if r= 2 or 3.

We can extend the problem to the case where there are 9 urns each of which have the required mixture of black and white balls. (Such as $p_1 = 0,1$, $p_2 = 0,2 \ldots\ldots p_q = 0.9$). The table becomes as follows:

r	0	1	2	3
P(r|1/10)	.729	.243	.027	.001
P(r|2/10)	.512	.384	.096	.008
P(r|3/10)	.343	.441	.189	.027
P(r|4/10)	.216	.432	.288	.064
P(r|5/10)	.125	.375	.375	.125
P(r|6/10)	.064	.288	.432	.216
P(r|7/10)	.027	.189	.441	.343
P(r|8/10)	.008	.096	.384	.512
P(r|9/10)	.001	.027	.243	.729

For a given possible value of r, we choose p which is most likely to have given rise to data. (Underlined ones).

(Example 2 ---- Continuous)

Suppose that the random variable Z has a continuous distribution with density function,

$$f(Z|\ \Theta)$$

where Θ is a single parameter which we want to estimate. The distribution of a sample of size n is

$$P\ (Z_1, Z_2 \ldots\ldots Z_n)\ \Pi dZ_i = \prod_{i=1}^{n} f\ (Z_i\ |\Theta)\ dZ_i. \qquad (2)$$

Once the Z_i's have been observed, the likelihood of Θ is defined by

$$\ell(\ \Theta|Z) = \prod_{i=1}^{n} f\ (Z_i|\Theta\ ) \qquad (3)$$

which is of the same form as P but in which Z's are fixed and Θ becomes variable.

The log likelihood of Θ is defined by

$$L(\Theta|Z) = \Sigma \log f(Z_i|\Theta). \tag{4}$$

The estimated Θ is the solution of $\frac{\partial \ell}{\partial \Theta} = 0$ or $\frac{\partial L}{\partial \Theta} = 0$. The reason why L is used is that the exponential form appears very often in statistics. If the normal distribution with unknown variance and mean is assumed, i.e.,

$$f(Z|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma}2} \; e^{-(Z-\mu)^2/2\sigma^2} \tag{5}$$

$$\ell(\mu,\sigma^2|Z) = (\sigma^2)^{-\frac{n}{2}} \exp - \sum_{i=1}^{n} (Z_i-\mu)^2/2\sigma^2 \tag{6}$$

$$L(\mu,\sigma^2|Z) = -\frac{n}{2} \log \sigma^2 - \sum_{i=1}^{n} \frac{(Z_i-\mu)^2}{2\sigma^2} \tag{7}$$

$$\frac{\partial L}{\partial \mu} = 0 \quad \rightarrow \frac{1}{\sigma 2} \sum (Z_i - \mu) = 0 \quad \therefore \; \mu = \bar{Z} \tag{8}$$

$$\frac{\partial L}{\partial \alpha 2} = 0 \quad \rightarrow \; -\frac{n}{2} \; \frac{1}{\sigma 2} + \frac{1}{2\sigma^4} \sum (Z_i \; \mu)^2 = 0 \quad \therefore \quad \hat{\sigma}^2 = \frac{\sum (\bar{Z} - Z_i)^2}{n} \tag{9}$$

In many examples, for moderate and large samples the log-likelihood function will be unimodal and can be adequately approximated over a sufficiently extensive region near the maximum by a quadratic function. The second derivatives in such cases provide measures of "spread" at its maximum of the likelihood function and can be used to calculate approximate standard deviations for the estimates.

4.2 Use of the Maximum Likelihood Functions to Estimate Parameters of Time Series

4.2.1

Now let us discuss how to use the maximum likelihood functions to estimate the parameters in ARIMA models. The idea is to calculate the residual terms by using the pre-estimated parameters and to use the maximum likelihood function for Gaussian distribution with pre-estimated variance. Suppose we have n + d observations. The ARIMA model (p,d,q) can be written as

$$a_t = \tilde{W}_t - \phi_1 \tilde{W}_{t-1} \cdots\cdots \phi_p \tilde{W}_{t-p} + \Theta_1 a_{t-1} + \Theta_2 a_{t-2} + \ldots + \Theta_q a_{t-q} \tag{10}$$

where $W_t = \nabla^d Z_t$, $\tilde{W}_t = W_t - E[W_t]$.

$(d > 0 \rightarrow E[W_t] = 0)$.

It can be shown (proof omitted, See [1]) that the log-likelihood function is given by

$$L(\phi,\Theta,\sigma_z) = f(\phi,\Theta) - n\ln\sigma - \frac{S(\phi,\Theta)}{2\sigma_a^2} \tag{11}$$

$S(\phi,\Theta) = \sum_{t=-\infty}^{n} [a_t \mid \Theta,\phi,W]^2$ is called the unconditional sum of squares function. $[a_t|\phi,\Theta,W]$ denotes the expectation of a_t conditional on ϕ,Θ and W. (See Appendix 3 of Chapter 2.)

$f(\phi,\Theta)$ is important only for small n and L is dominated by the last term.

Therefore, the parameter estimates obtained by minimizing the sum of squares, least square estimates usually provide very good approximation to the maximum likelihood estimates. The unconditional sum of squares are calculated by computing $[a_t]$'s recursively.

4.2.2 General procedure for the unconditional sum of squares

Suppose that the W_t's are generated by the stationary forward model

$$\phi(B)\,\tilde{W}_t = \Theta(B)\,a_t \tag{12}$$

where $\nabla^d Z_t = W_t$ and $\tilde{W}_t = W_t - \mu$ (μ=0). The corresponding backward formula is

$$\phi(F)\,\tilde{W}_t = \Theta(F)\,a_t. \tag{13}$$

1) From (12) and (13) we obtain the dual set of equations

$$\phi(B)\,[\tilde{W}_t] = \Theta(B)\,[a_t] \tag{12'}$$

$$\phi(F)\,[W_t] = \Theta(F)\,[e_t] \tag{13'}$$

We first compute $[e_t]$'s by (13'). Several e_t's beyond e_n are assumed to be zero to start off the sequence. $[e_n]$, $[e_{n-1}]$.....$[e_1]$ are thus obtained. $[e_o]$, $[e_{-1}]$.... are zero because they are distributed independent of W.

2) We then calculate $[W_{-j}]$ by using $[W_j]$'s and $[e_t]$'s. Since there are autoregressive operators ϕ_i's $[W_{-j}]$ are extended to infinity although $[e_{-j}]$'s vanish. But the condition of stationarity ensures that $[W_{-j}]$'s decrease and becomes essentially zero beyond some point.

3) After $[W_{-j}]$'s vanish, we can compute the forward sequence $[a_j]$'s by (12'). $[a_{-j}]$'s = 0 beyond the point where $[W_{-j}]$'s $\simeq$ 0. Thus $[a_j]$'s are obtained beyond n.

4) Compute $S(\phi,\Theta) = \sum [a_t]^2$.

An illustrative example is given below to understand the general procedure. In table (4.1) $[W_t]$ is the given data. An assumed model is

$$(1-0.3B)\ W_t = (1-0.7)\ a_t.$$

Then, (12´)and(13´) can be written as

$$[a_t] = [W_t] - 0.3\ [W_{t-1}] + 0.7\ [a_{t+1}], \text{ and}$$

$$[e_t] = [W_t] - 0.3\ [W_{t-1}] + 0.7\ [e_{t+1}] \text{ respectively.}$$

where $[W_t] = W_t$. $(t = 1,2,....n)$

TABLE 4.1

t		$[a_t]$	$0.7\ [a_{t-1}]-0.3[W_{t-1}]$	$[W_t]$	$-0.3[W_{t+1}]$	$0.7[e_{t+1}]$	$[e_t]$
-4						0	0
-3		-0.01	0.00	-0.01	(2) 0.01	0	0
-2		-0.04	-0.01	-0.03	0.03	0	0
-1		-0.11	-0.03	-0.09	0.09	0	0
0		-0.36 -1.20	-0.08 -0.25	-0.31 -1.04	0.31 0.60	1.64	0
1		1.47	-0.84	0.31	2.0	-0.24	0.58	2.34
2		1.23	1.03	-0.60	0.8	0.09	-0.06	0.83
3	(3)	0.32	0.86	-0.24	-0.3	0.09	0.13	-0.08
4		0.02	0.23	0.09	-0.3	0.57	-0.09	0.18
5		-1.80	0.01	0.09	-1.9	-0.09	1.86	-0.13
6		-0.39	-1.26	0.57	0.3	-0.96	3.32	2.66
7		2.84	-0.27	-0.09	3.2	-0.48	2.02	4.74
8		2.63	1.99	-0.96	1.6	0.21	1.08	2.89
9		0.66	1.84	-0.48	-0.7	-0.90	3.14	1.54
10		3.67	0.46	0.21	3.0	-1.29	2.78	4.49
11		5.97	2.57	-0.90	4.3	-0.33	0	3.37
12		3.99	4.18	-1.29	1.1			(1)

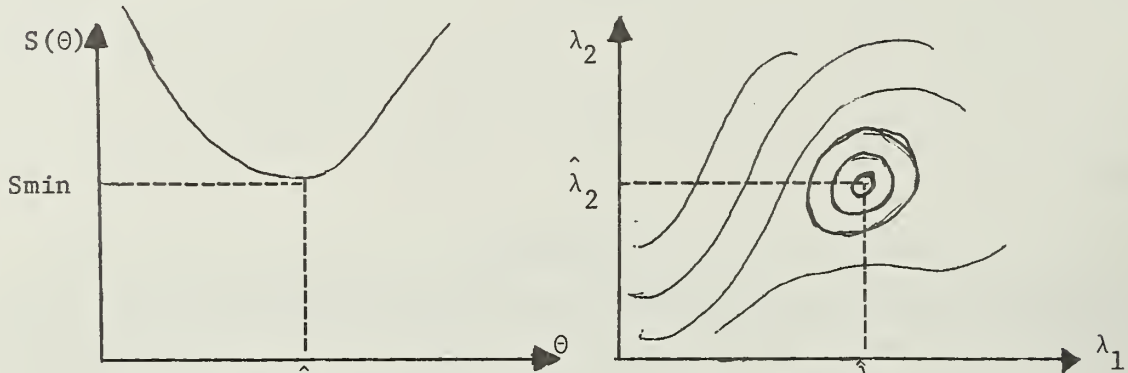
( ) initially given or assumed

$$S(0.3,\ 0.7) = \sum_{t=-4}^{12} [a_t]^2 = 89.2$$

We can perform the second iterative cycle, but in most of the cases it is not necessary; the convergence is very rapid. In this example, the second iteration yields $S = \sum [e_t]^2 = 89.3$.

4.3 Graphical Study of the sum of squares functions

It is very helpful to use some graphical displays to determine the region of the global minimum of the sum of squares functions.



1 - parameter case　　2 - parameter case

If S is a function of two parameters, a contour map or a "cross section graph" are very useful to understand the characteristics of S.

Some Remarks

(1) We must examine the total characteristics of the maximum likelihood functions. We should note that there are more than one minimum in some cases.

(2) It is possible to detect overfitted case by checking whether the minimum is located on the boundary of the parameter space.

(3) Sometimes there are constraints on the domain of parameter space. In such cases, the maximum likelihood function must be minimized within the limited domain.

4.4 Variances, Covariances and Confidence Region

Near the minimum point, the maximum likelihood function can be expanded by Taylor's theorem:

$$L(\xi) = L(\beta,\sigma_a^2) \simeq L(\hat{\beta},\sigma_a^2) + \frac{1}{2}\sum_{i=1}^{k}\sum_{i=1}^{k} \ell_{ij}(\beta_i-\beta_i)(\beta_j-\beta_j) + \cdots\cdots \quad (14)$$

where $\ell_{ij} = \dfrac{\partial^2 L}{\partial\beta_i\beta_j}$, k = p+q (the number of parameters)

The variance matrix is then defined as:

$$V(\hat{\beta}) = \{\cdot E[\ell_{ij}]\}^{-1} = 2\,\sigma_a^{\,2}\,\{S_{ij}\}^{-1} \qquad (15)$$

where

$$S_{ij} = \frac{\partial^2 S(\beta|W)}{\partial\,\beta_i\,\partial\beta_j} \qquad (k \times k \text{ matrix})$$

(It is assumed that the size of samples are moderate and large so that S is the dominant part of L in (14). It can be shown that

$$\hat{\sigma}_a^{\,2} \simeq \frac{S(\hat{\beta})}{n} \qquad \text{(proof omitted, see [1]).} \qquad (16)$$

Furthermore, the confidence region can be determined for a given confidence level ε in the following way:

$$-\sum E[\ell_{ij}]\,(\beta_i - \hat{\beta}_i)\,(\beta_j - \hat{\beta}_j) < \chi_\varepsilon^{\,2}(k) \qquad (17)$$

where $\chi_\varepsilon^{\,2}(k)$ is χ^2 - distribution function with degree of freedom k. Using the above relation, the confidence region for a given confidence level ε is

$$S(\beta) \le S(\hat{\beta})\quad\left(1 + \frac{\chi_\varepsilon^{\,2}(k)}{n}\right). \qquad (18)$$

4.5 On Parameter Redundancy

Theoretically, the model

$\phi(B)\,\tilde{W}_t = \Theta(B)\,a_t$

is identical with the model

$$(1 - \alpha B)\,\phi(B)\,\tilde{W}_t = (1-\alpha B)\,\Theta(B)\,a_t.$$

But in the estimation procedure this redundancy causes serious difficulties. Therefore we should avoid the situation where redundancy or near-redundancy occurs. Let us take an example in ARMA (2,1) model:

$$(1-1.3B+0.4B^2)\,\tilde{W}_t = (1-.5B)\,a_t.$$

$$\text{or } (1-.5B)(1-.8B)\,\tilde{W}_t = (1-.5B)\,a_t.$$

which is identical with

$$(1-.8B)\,\tilde{W}_t = a_t.$$

A near redundant case can be, for example

$$(1-.4B)\,(1-.8B)\,\tilde{W}_t = (1-.5B)\,a_t.$$

There are more than one combination of parameters which yield similar $[a_t]$'s and consequently similar likelihoods can be found; a change of parameter value on the left hand side can be nearly compensated by a suitable change on the right hand side to give the same likelihoods.

The sum of squares contour surfaces in the 3-dimensional parameter space will be cylinders rather than ellipsoids and a line of near least squares solutions will be obtained rather than a clearly defined point minimum.

The identification procedure in Chapter 3 is very important in the sense that it enables us to avoid such disastrous situations. Here is an example in which the direct cancellation of factors occur in ARMA (1,1) process:

$$(1-\phi B)\,\tilde{W}_t = (1-\Theta B)\,a_t.$$

If $\phi = \Theta$ then $\tilde{W}_t = a_t$,

that is, $\tilde{W}_t$ becomes a white noise process itself. The sum of squares function is constant on $\Theta = \phi$. In practice the identification technique will reveal any such situation easily. Note the following things:

(1) We should avoid mixed processes containing near common factors and we should be alert to the difficulties that can result.

(2) We will automatically avoid such processes if we use identification and estimation procedures intelligently.

4.6 Description of Program MLES

Input parameters	W_t, (time series)
	σ_a^2, Θ, ϕ (pre-estimated parameters)
Procedure	RESIDUAL (residual terms)
Output	S (least square function)

This program computes the sum of the squares function from given data using the pre-estimated parameters. First the residual terms a_t are reproduced and then the least squares function is computed by summing up a_t^2's.

Chapter 5 Nonlinear Estimation

In the method described in Chapter 4, we must compute the sum of the squares function $S(\beta)$ many times in the parameter space around the pre-estimated parameters β_0, to locate the point which gives the minimum of $S(\beta)$, i.e. $S(\hat{\beta})$. This is very time consuming when there are many parameters to be estimated, although it gives global behavior of the function $S(\beta)$ and is less dangerous. We can improve this by using some nonlinear methods. The classical Gauss-Newton's method is introduced in Section 1 and its modification by Marquardt is discussed in Section 2. Computation of the derivatives of $S(\beta)$ is discussed in Section 3. Comparison of two methods is also given.

5.1 Linearization of the models (Gauss-Newton's method)

Let us take an ARMA (p,q) model

$$\phi(B)\ \tilde{\omega}_t = \Theta(B)\ a_t \qquad t = 0,1,2,\ldots,n.$$

Let $\vec{\beta}$ denote θ's and ϕ's. It is a $(p + q)$ dimensional vector. (Let us put $k = p + q$.) We can regard $[a_t]$'s as continuous functions of $\vec{\beta}$ and expand them in Taylor's series up to the first order around a given set of initial values $\vec{\beta}_0$.

$$[a_t] = [a_{t_0}] + \sum_{i=1}^{k} (\beta_i - \beta_{i0}) \frac{\partial a_t}{\partial \beta_i} + 0\left((\beta_i - \beta_{i0})^2\right) \qquad (1)$$

where

$$[a_{t_0}] = [a_t \mid w, \vec{\beta}_0].$$

In matrix notation, (1) is written as

$$[\vec{a}] = - X (\vec{\beta} - \vec{\beta}_0) + [a_0]$$

where

$[\vec{a}]$ and $[a_0]$ are N dimensional vectors and $x_{ij} = - \dfrac{\partial a_i}{\partial \beta_j}$.

Then $S(\vec{\beta})$ is computed as follows:

$$S(\vec{\beta}) = \sum_{t=1}^{N} a_t\ (\vec{\beta})^2 = [\vec{a}]^t\ .\ [a]$$

$$= [\vec{a}_0]^t .\ [a_0] - 2\ [\vec{a}_0]^t\ X\ (\vec{\beta} - \vec{\beta}_0) + (\vec{\beta} - \vec{\beta}_0)\ X^t\ X(\vec{\beta} - \vec{\beta}_0).$$

By putting $\frac{\partial S}{\partial \beta_j} = 0$, we obtain a system of equations

$$X^t X(\vec{\beta} - \vec{\beta}_0) = X [\vec{a}_0] \tag{2}$$

or

$$\sum_{t=0}^{N} \sum_{j=1}^{k} X_{t,i} X_{t,j} (\beta_j - \beta_{j0}) = \sum_{t=0}^{N} X_{t,j} a_{t0} \tag{2'}$$

$i = 1,2,\ldots,k.$

By putting $P_{ij} = \sum_{t=0}^{N} X_{t,i} X_{t,j}$ and $g_i = \sum_{t=0}^{N} X_{t,j} a_{t0}$ we can write (2) as

$$P(\delta\vec{\beta}) = \vec{g} \tag{3}$$

where $\delta\vec{\beta} = \vec{\beta} - \vec{\beta}_0$ is the correction to the parameters. Therefore we can obtain the improved values for β by solving (3) with regard to $\delta\vec{\beta}$.

(3) is usually solved after scaling, [5]. That is,

$$P^* = (p_{ij}^*) = \frac{p_{ij}}{\sqrt{p_{ii}}\ \sqrt{p_{jj}}}$$

$$\vec{g}^* = (g_i^*) = \frac{g_i}{\sqrt{p_{ii}}}$$

Then a new system of equations

$$P^*\ \vec{\delta\beta}^* = \vec{g}^*. \tag{3'}$$

is solved exactly (by Gauss elimination). $\delta\vec{\beta}$ in (3) is obtained from $\delta\vec{\beta}^*$.

$$\delta\vec{\beta} = \begin{pmatrix} \sqrt{p_{11}} & & & \\ & \sqrt{p_{22}} & & \\ & & \ddots & \\ & & & \sqrt{p_{kk}} \end{pmatrix}^{-1} \delta\vec{\beta}^*$$

or $\delta\beta_i = \delta\beta_i^* / \sqrt{p_{ii}}$

Since the Taylor's expansion (1) is correct only near $\vec{\beta}_0$ (it gives only the tangential plane of $S(\vec{\beta})$ at $\vec{\beta} = \vec{\beta}_0$), a single adjustment will not immediately produce the least square values; several iterations are necessary. Naturally convergence is faster if reasonably good initial guesses are used. This is why we have been bothered so much in making good preliminary estimation of parameters in 3.3.

5.2 A Modification by Marquardt

Sometimes Gauss-Newton's method turns out to be very unstable. Marquardt improved its convergence property by increasing all the eigenvalues of P* in (3′) by a fixed positive amount [6]. The theory of his method is not shown here, but the algorithm is briefly discussed.

After obtaining the scaled system (3′), the matrix P* is changed so that

$$P_M = P^* + \lambda I$$

where λ is a positive number. Then

$$P_M \ \delta\vec{\beta}^* = \vec{g}^* \tag{4}$$

is solved. It can be proved that the new parameters $\vec{\beta}' = \vec{\beta}_0 + \delta\vec{\beta}$ will lead to a new sum of squares $S(\vec{\beta})$, which is always smaller than $S(\beta)$ for a sufficiently large λ, unless $\vec{\beta}'$ is already at a minimum of $S(\vec{\beta})$. Our strategy to find λ is as follows:

(1) If $S(\beta') < S(\beta)$, the parameter correction $\delta\vec{\beta}$ is tested. If $\max |\delta\beta_i| < \varepsilon$ where ε is some prescribed small number such as 10^{-8}, convergence is assumed. Otherwise, $\vec{\beta}$ is modified and λ is multiplied by $\tau (\tau < 1)$ and computation continues with new parameters. (i.e. new P and $\vec{g}$ are calculated).

(2) If $S(\beta') > S(\beta)$, λ is divided by τ (note that $\lambda/\tau > \lambda$) and computation is continued to solve the system of linear equation (4) with new λ until reduced sum of squares $S(\vec{\beta})$ is found or λ becomes unreasonably large. It is suggested by Marquardt [6] and Bard [7] to put $\lambda = .01$ initially.

5.3 Numerical estimates of the derivatives

It is necessary to calculate the derivatives $X_{i,j}$ to construct the system of equations (2). There are two methods; the direct method and the numerical method. The former is to make use of some recursive formulae and will not be discussed in this document. The latter is essentially the same as deriving residual terms in 4.2.2. We calculate both

$$[a_t | \beta_1, \beta_2, \ldots \beta_k] \text{ and } [a_t | \beta_1, \beta_2, \ldots \beta_i + \delta\beta_i, \ldots \beta_k]$$

and then obtain $X_{t,j}$ by

$$X_{t,j} = \frac{[a_t | \beta_1, \beta_2, \ldots \beta_i + \delta\beta_i, \ldots \beta_k] - [a_t | \beta_1, \beta_2, \ldots \beta_k]}{\delta\beta_i} .$$

This method has the following advantages:

(1) applicability is universal.

(2) the same subprogram (RESIDUAL) as in 4.2.2 can be used.

5.4 Comparison of the two methods

Both methods have been implemented and tested. With the test example given in Appendix (197 data) Gauss-Newton's method turned out to be unstable; oscillation was observed. Marquardt's method, on the other hand, was stable and converged very quickly. (Table 5-1).

TABLE 5-1 197 Data (See the appendix)

	Values of S(β)			
model / iteration	(1, 0, 1)		(0, 1, 1)	
	N - G	Marq.	N - G	Marq.
1	19.264	19.260	322.07	320.76
2	19.255	19.254	318.35	318.35
3	19.254	conv.	318.36	318.32
4	19.254		318.36	318.31
5	19.254		318.37	conv.
6	19.254		318.37	
7	conv.		318.37	
8			318.38	
9			318.38	
10			318.39	
11			318.38	
12			318.40	
13			318.39	
14			318.40	
15			318.39	
16			318.41	
17			318.39	
18			318.41	
19			318.39	
20			318.41	

Note: Convergence is checked by max $|\delta\beta_i|$, not by $\delta S(\beta)$.

5.5 Description of program MLQT

Input	W	time series
	N	size of ω
	ϕ	preestimated order and parameters
	θ	
	P	
	Q	
	mean	$\bar{\omega}$
	ε	criterion for convergence of
	δ	used to calculate the derivatives
	I max	maximum number of iterations
	τ	explained in 5.2
	λ	
	ℓ max	a bound for λ .
Output	S	sum of the squares
	$\hat{\phi}$	parameters estimated
	$\hat{\theta}$	
	X	variance covariance matrix (to be added)

Subprograms

GAUSSLU

RESIDUAL

Functions

1. Computes derivatives using RESIDUAL.
2. Computes improved values of parameters using GAUSSLU. (Marquardt's algorithm.)
3. Computes $S = \Sigma a_t^2$ and X (to be added)

Chapter 6 Model Diagnosis

Having estimated the parameters, we must examine whether or not the model which we have built really fits the given data. One way of checking the models is called a method of overfitting, that is, to estimate the parameters in a model somewhat more general than that which we believe to be true. This method assumes our knowledge on the direction in which the model is likely to be inadequate. Therefore it is not general and will not be discussed here. The second way of checking the models is to make use of the white noise term which we have computed to obtain the sum of the squares function S; to check whether a_t's are sampled from Gaussian population. There are two methods; that is, the one which employes (1) the autocorrelation functions and (2) the cumulative periodogram.

6.1 Autocorrelation Check

Suppose a model

$\phi(B)\ W_t = \Theta(B)\ a_t$ with $W_t = \nabla^d Z_t$ has been identified and the maximum likelihood estimates ($\hat{\phi}$, $\hat{\Theta}$) obtained for the parameters.

As in Chapter 5, the quantities $\hat{a}_t = \hat{\Theta}^{-1}(B)\ \hat{\phi}(B)\ \tilde{W}_t$. $t=1,2,\ldots,N$ are supposed to have Gaussian distribution with mean 0 and the variance σ_a^2. Therefore it is to be expected that study of the a_t's indicates the adequacy of the model.

It is known in χ^2- theory that the quantity

$$\sum(\hat{a}_t - \bar{\hat{a}}_t)^2 / \sigma_a^2 \qquad (1)$$

where $\bar{\hat{a}}_t = \frac{1}{n}\sum \hat{a}_t$ is distributed as $\chi^2(n-1)$, if a_t's are sampled from $N(0,\sigma_a^2)$. Therefore, if n is sufficiently large, then the quantity (1) should be approximately N-1 and the χ^2 testing with some confidence level is available. In practice, however, this criterion is too crude.

Hence, more sophisticated diagnoses are necessary. How about using the autocorrelations r(k) obtained from a_t's? If a_t's are white noise, the autocorrelations r(k) $k \geq 1$ will be normally distributed around 0 with standard deviation $\frac{\sim 1}{\sqrt{n}}$. Unfortunately, this diagnosis is not accurate enough in

practice, either; unless the lag k is sufficiently large, this criterion is underestimated and dangerous to use.

The best diagnosis to exploit the autocorrelation functions is called "portmanteau" test in which $r_k(a)$'s are taken as a whole to test inadequacy of the model rather than individually. Suppose that we have the first K (say, 20) autocorrelations $r_k(\hat{a})$ from an ARIMA (p,d,q), then it is possible to show [8] that, if $\hat{a}_t$'s are sampled from Gaussian population the quantity

$$Q = n \sum_{k=1}^{k} r_k^2 (\hat{a})$$

is approximately distributed as $\chi^2(k-p-q)$, where $n = N-d$ is the number of W's used to fit the model. Hence, we can examine χ^2-testing with regard to Q.

6.2 Cumulative Periodogram Check

This is the only section in this document where the power spectrum (periodogram) is used in practice. We have assumed in the beginning that there is no seasonal or periodic nature in the data. But in actual cases there will be many data which involve some periodic characteristics. Therefore we discuss in this section a way of detecting periodicity in a_t's. The autocorrelation function does not indicate sensitively such periodic nature. The power spectrum, on the other hand,is very suitable to detect periodic patterns buried in white noise.

The periodogram is defined in A2 of Chapter 2 as

$$I(f_i) = \frac{2}{n} \left[\left(\sum_{t=1}^{n} a_t \cos 2\pi f_i t \right)^2 + \left(\sum_{t=1}^{n} a_t \sin 2\pi f_i t \right)^2 \right]$$

where $f_i = i/n$ is the frequency.

The power spectrum p(f) has a constant value $2\sigma_a^2$ over the frequency domain 0=0.5 cycle if a_t's are sampled from $N(0, \sigma_a^2)$. Consequently the cumulative spectrum $P(f) = \int_0^f p(g)\, dg$ is a straight line if plotted against f $(0 \leq f \leq 0.5)$, or

$P(f)/\sigma_a^2$ is a straight line in $0 \le f \le 0.5$.

$P(f)/\sigma_a^2$ can be estimated by

$$C(f_i) = \frac{\sum_{j=1}^{i} I(f_j)}{ns^2}$$

where s^2 is an estimated variance of a_t's, i.e. σ_a^2.

We shall refer to $C(f_i)$ as the normalized cumulative periodogram. The reason why the cumulative spectrum is used rather than the spectrum itself is that the Kolmogorov-Smirnov testing is available to give limits with the confidence levels around straight lines.

Kolmogorov-Smirnov testing [9] can be described as follows. Let $F(x)$ denote the theoretical cumulative distribution function of a random variable (continuous) x, and $F_q(x)$ an empirical cumulative distribution obtained from q samples. Then $D_q = \max |F_q(x) - F(x)|$ takes values greater than or equal to $\frac{\lambda}{\sqrt{q}}$ with probability $K(\lambda) = 1-2 \sum_{\nu=1}^{\infty} (-1)^{\nu-1} e^{-2(\nu\lambda)^2}$ when $n \to \infty$. Therefore with a given confidence level $1-\varepsilon$, $F_n(x)$ should be confined within a strip $F(x) \pm \frac{K(\varepsilon)}{\sqrt{q}}$.

As for the case in this section $F(x) = x$ $(0 \le x \le .5)$ and $q = \frac{n-2}{2}$ (n even) or $q = \frac{n-1}{2}$ (n odd) (∵ Frequency domain is (0, .5)). Therefore the limit lines are drawn at distances $K(\varepsilon)/\sqrt{q}$ above and below the line $F(x) = x$. Some values of $k(\varepsilon)$ are given in Table 6.1.

ε	.01	.05	.10	.25
$K(\varepsilon)$	1.63	1.36	1.22	1.02

TABLE 6.1.

6.3 Correction of Model Parameters

In some cases the order of an identified model is correct but the parameters are not fitted well. It is possible to find closer values for parameters by analyzing a_t's.

Suppose that the correct model is

$$\phi(B)\ \nabla^d Z_t = \theta(B)\ a_t$$

but that an incorrect model

$$\phi_0(B)\ \nabla^d Z_t = \theta_0(B)\ b_t$$

is used. Then b_t will not be Gaussian any more.

$$b_t = \theta_o^{-1}(B)\ \phi_0(B)\ \nabla^d Z_t = \theta_o^{-1}(B)\ \phi_o(B)\ \phi^{-1}(B)\ \theta(B)\ a_t\ .$$

That is, b_t itself is an ARMA process. Therefore, by identifying b_t in the methods described in Chapters 3 and 4, for example, $\phi'(B)\ \nabla^{d'} b_t = \theta'(B)\ a_t$. we can obtain a new improved model

$$\phi'(B)\ \phi_0(B)\ \nabla^{d+d'} Z_t = \theta_0(B)\ \theta'(B)\ a_t.$$

List of References

[1] Box, G.E.P., and Jenkins, G.M., Time Series Analysis, Holden-Day Inc. 1970.

[2] Bartlette, M.S., Jour. Roy. Stat. Soc. B8, (1946), p. 27.

[3] Blackman, R.B., and Tukey, J.W., The Measurement of Power Spectra, Dover 1958.

[4] Wilson, G. J., J.S.I.A.M. (Num. Anal.) 6 , (1969), p. 1.

[5] Curry, H.B., Quart. Appl. Math. 2 (1944) pp. 258-261.

[6] Marquardt, D.W., J.S.I.A.M., vol. 11, no. 2 (1963), pp. 431-441.

[7] Bard, Y. J.S.I.A.M., (Num. Anal.), vol. 7, no. 1 (1970), pp. 157-186.

[8] Box, G.E.P., and Pierce, E.A., Jour. Amer. Stat. Soc. 64 (1970).

[9] Schreider, Ed. Y. A., The Monte Carlo Method, Pergamon (1966) or Knuth, D.E., The Art of Computer Programming, Vol. 2, Addison Wesley (1969).

[10] Quenouille, M. H., Jour. Royal Stat. Soc. B11, 68 (1949).

APPENDIX 1

SOURCE PROGRAM

```
THIS FILE PRINTED ON: 02/13/72  AT: 05:19:53.

    BEGIN
        FILE CARD(KIND=0,TITLE="NUMBERS.");
        FILE   LINE(KIND=7,MAXRECSIZE=22);
       INTEGER    I,J,K,P,Q,N,D,IM;
       REAL          SIGMA,VAR,MEAN,EPSILON,DLT,GAMA,ZETA;
       REAL        ARRAY  ATCR,THETA,PHI[0:20];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          PROCEDURE   USID(Z,N,MEAN,ATCR,VAR,CD,L,D,W);
               VALUE  N,CD,L,D;
               INTEGER D,CD,L,N;
               REAL    MEAN,VAR;
               REAL ARRAY   Z,W[1],ATCR[0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                 BEGIN
             %
             %
             %           PARAMETERS
             %      Z     :TIME SERIES TO BE ANALYZED (ORIGINAL)
             %      W     :TIME SERIES (DIFFERENCED)
             %      N     :NUMBER OF DATA POINTS(I.E. Z)
             %      MEAN  :AVERAGE OF Z  (COMPUTED IN THIS PROCEDURE)
             %      VAR   :VARIANCE OF Z (COMPUTED IN THIS PROCEDURE)
             %      ATCR  :AUTOCORRELATION FUNCTION (COMPUTED HERE)
             %      CD    :MAXIMUM LAG OF ATCR
             %      D     :ORDER OF HIGHER DIFFERENCE
             %      L     :MAXIMUM LAG OF PACF
      %
             %      THIS PROCEDURE IS FOR THE INITIAL ESTIMATES OF AUTO-
             %   REGRESSIVE AND MOVING-AVERAGE MODELS, THE ORDER IS P OR Q,
             %   RESPECTIVELY.THOSE INTIAL ESTIMATES ARE USED IN THE MAXIMUM-
             %   LIKELIHOOD ESTIMATES.
             %
             %
             %
             %
             %
        FORMAT OUT
        F11(X12,"ORDER OF THE DIFFERENCE = ",I2/),
        F17(X12,"NUMBER OF DATA=",X2,I4),
         F12(X12,"MEAN VALUE = ",X1, R11.4),
        F14(//X5,"ORDER",X3,"AUTOCORRELATIONS",X3,
            "PARTIAL AUTOCORRELATIONS"//),
        F16(X12,"VARIANCE = ",X1,R11.4),
          F15(X7,I2,X9,R11.4,X14,R11.4);
                    REAL    SD;
                    DOUBLE    SUM,SM;
                    REAL ARRAY          SQ,PACC[0:L];
                    DOUBLE ARRAY PACF[1:L,1:L];
                    INTEGER  I,J,K,M,NN,INDX,JNDX,Q;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        PROCEDURE    CORRELOPRINT(ATCR,N,W);
          VALUE  N,W;
          INTEGER    N,W;
          REAL ARRAY   ATCR[0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
              BEGIN
                    FORMAT OUT
                    F51(X20,"AUTO CORRELATION FUNCTION"//),
```

```
            F52(X20,"V",10(9(",","V")),
            F53(X20,"A",10(9(",","A")),
            F54(X3,R11,4,X2,I2,X1,"I",X50,"0",*("X")),
            F55 (X3,R11,4,X2,I2,X1,"I",X*,*("X"),"0"),
            F56(X3,R11,4,X2,I2,X1,"I",X50,"0"),
            F57(X18,"-1,0",X7,"-,8",X7,"-,6",X7,"-,4",X7,"-,2",
                X8,"0",X8,",2",X8,",4",X8,",6",X8,",8",X7,"1,0"
                ),
            F58(X20,"PARTIAL AUTOCORRELATION FUNCTION"//),
            F59(X10,"Q",X5,"ERROR BOUND"/),
            F60(X9,I2,X5,R11,4);
            INTEGER   I,J,K;
            REAL   TEMP;
                 WRITE(LINE[SKIP 1]);
                 IF W=0 THEN
                      WRITE(LINE,F51)
                 ELSE
                      WRITE(LINE,F58);
                 WRITE(LINE[SPACE 2]);
                 WRITE(LINE,F57);
                 WRITE(LINE,F52);
                FOR I:=0 STEP 1 UNTIL N DO
                   BEGIN
                     TEMP:=ATCR[I];
                     K:=50+TEMP;
                     IF K>0 THEN
                     WRITE(LINE,F54,TEMP,I,K)
                     ELSE
                     IF K<0 THEN
                     WRITE(LINE,F55,TEMP,I,50+K,-K)
                     ELSE
                     WRITE(LINE,F56,TEMP,I);
                   END;
                  WRITE(LINE,F53);
                 WRITE(LINE,F57);
           WRITE(LINE[SPACE  2]);
           IF W=0 THEN
             BEGIN
                WRITE(LINE,<X10,"IF THE PROCESS IS MA(Q",
                     "),THEN THE AUTOCORRELATIONS HIGHER "
                     "THAN  Q    ARE WITHIN THE RANGE OF:"//>
                     );
                WRITE(LINE,F59);
                FOR I:=1 STEP 1 UNTIL L DO
                     WRITE(LINE,F60,I,SQ[I]);
             END   ELSE
             BEGIN
               WRITE(LINE,<"IF THE PROCESS IS AR(P),THE ",
                     "PARTIAL AUTOCORRELATIONS HIGHER THAN",
                     " P ARE WITHIN THE RANGE OF:",X3,R11,4//>
                     ,SD);
             END;
        END;  %%  CORRELOPRINT %%
COMMENT  THIS PROCEDURE IS TO CALCULATE THE MEAN VALUE,
       AUTOCORRELATIONS AND PARTIAL A.C. PAC IS COMPUTED IN
       DOUBLE PRECISION TO GET ENOUGH ACCURACY. ;
          WRITE(LINE[SKIP 1]);
          FOR I:=1 STEP 1 UNTIL N DO
               W[I]:=Z[I];
COMMENT          HIGHER DIFFERENCE;
```

```
          NN:=N;
          FOR J:=1 STEP 1 UNTIL  D DO
            BEGIN
               NN:=NN-1;
               FOR I:=1 STEP 1 UNTIL NN  DO
                 BEGIN
                    W[I]:=W[I+1]-W[I];
                 END;%SERIES
            END; %%ORDER
COMMENT     AVERAGE;
          SUM:=0;
          FOR I:=1 STEP 1 UNTIL NN DO
            SUM:=SUM+W[I];
          MEAN:=SUM/NN;
COMMENT     COVARIANCE;
          FOR K:=0 STEP 1 UNTIL CD DO
            BEGIN
               SUM:=0;
               FOR I:=1 STEP 1 UNTIL NN-K DO
                    SUM:=SUM+(W[I]-MEAN)*(W[I+K]-MEAN);
               IF K NEQ 0 THEN
               ATCR[K]:=SUM/VAR        ELSE
                 BEGIN
                   VAR:=SUM;
                   ATCR[0]:=1;
                 END;
            END;
          VAR:=VAR/NN;
COMMENT     PARTIAL AUTO CORRELATION FUNCTION;
          PACC[1]:=ATCR[1];
          PACF[1,1]:=DOUBLE(ATCR[1]);
          FOR J:=2 STEP 1 UNTIL L DO
            BEGIN
COMMENT        PHI(L,L);
               SUM:=SM:=0;
               FOR M:=1 STEP 1 UNTIL  J-1 DO
                  BEGIN
                    SUM:=SUM+PACF[J-1,M] MUX ATCR[J-M];
                    SM:=SM+PACF[J-1,M] MUX ATCR[M];
                  END;
               PACF[J,J]:=(DOUBLE(ATCR[J])-SUM)/(1-SM);
               PACC[J]:=REAL(PACF[J,J]);
COMMENT        PHI(L,K)           ;
               FOR K:=1 STEP 1 UNTIL J-1 DO
                    PACF[J,K]:=PACF[J-1,K]-PACF[J,J]*
                               PACF[J-1,J-K];
            END;
COMMENT     TO TEST THE ASSUMPTION THAT THE PROCESS IS MA(Q);
          SD:=0;
          FOR K:=1 STEP 1 UNTIL L DO
            BEGIN
               SD:=SD+ATCR[K]**2;
               SQ[K]:=SQRT((1+SD+SD)/NN);
            END;
COMMENT     THE STANDARD DEVIATION SD IS CALCULATED;
COMMENT     TO TEST THE ASSUMPTION THAT THE PROCESS IS AR(P),
          THE STANDARD DEVIATION SD IS CALCULATED;
          SD:=1/(SQRT(NN));
          WRITE(LINE[SKIP 1]);
          WRITE(LINE,</X10,"MESSAGES FROM US!D"//>);
```

```
              WRITE(LINE,F17,N);
             WRITE(LINE,F11,D);
             WRITE(LINE,F12,MEAN);
             WRITE(LINE,F16,VAR);
             WRITE(LINE,F14);
             WRITE(LINE,F15,0,ATCR[0],PACC[0]);
          FOR K:=1 STEP 1 UNTIL L DO
             WRITE(LINE,F15,K,ATCR[K],PACF[K,K]);
            CORRELOPRINT(ATCR,CD,0);
            CORRELOPRINT(PACC,L,1);
             WRITE(LINE[SKIP 1]);
          END  PROCEDURE;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%?
      PROCEDURE GAUSSLU(N,A,B,X);
           VALUE     N;
           INTEGER   N;
           REAL ARRAY A[1,1],B,X[1];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%C%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%?
           BEGIN
           FORMAT OUT
                F1(X10,"MATRIX WITH ZERO ROW IN DECOMPOSE"/),
                F2(X10,"SINGULAR MATRIX IN DECOMPOSE."/);
             INTEGER ARRAY      PS[1:N];
             REAL  ARRAY        LU[1:N,1:N];
    FORMAT OUT  FF1(X10,R11.4,X10,R11.4,X10,R11.4);
      PROCEDURE SINGULAR(WHY);
           VALUE     WHY;
           INTEGER   WHY;
           BEGIN
                IF WHY=0 THEN
                     WRITE(LINE,F1);
                IF WHY=1 THEN
                     WRITE(LINE,F2);
           END; %% SINGULAR %%
      PROCEDURE DECOMPOSE(N);  VALUE  N;INTEGER N;
           COMMENT   COMPUTES TRIANGULAR MATRICES L AND U AND
                     PERMUTATION MATRIX P SO THAT LU=PA.STORES
                     L-I AND U IN LU.ARRAY PS STORES PERMUTED
                     ROW INDICES;
           BEGIN
             REAL ARRAY SCALES[1:N];
             INTEGER   I,J,K,PVI;
             REAL       NR,PV,SIZE,MAX,MULT;
             LABEL   EL;
             COMMENT   INITIALIZE PS,LU AND SCALES;
                FOR I:=1 STEP 1 UNTIL N DO
                   BEGIN
                     PS[I]:=I;
                     NR:=0;
                     FOR J:=1 STEP 1 UNTIL N DO
                       BEGIN
                          LU[I,J]:=A[I,J];
                          IF NR<ABS(LU[I,J])  THEN
                          NR:=ABS(LU[I,J]);
                       END;
                     IF NR NEQ 0 THEN
                     SCALES[I]:=1/NR ELSE
                       BEGIN
                          SCALES[I]:=0;
                          SINGULAR(0);
```

```
            END;
        END;
        COMMENT   GAUSSIAN ELIMINATION WITH
                  PARITIAL PIVOTING;
        FOR K:=1 STEP 1 UNTIL N-1 DO
           BEGIN
             MAX:=0;
             FOR I:=K STEP 1 UNTIL N DO
                BEGIN
                  SIZE:=ABS(LU[PS[I],K])*SCALES[PS[I]];
             IF MAX<SIZE THEN
                     BEGIN
                       MAX:=SIZE;
                       PVI:=I
                     END;
                END;
             IF MAX=0 THEN
                BEGIN
                  SINGULAR(1);
                  GO TO EL;
                END;
             IF PVI NEQ K THEN
                BEGIN
                  J:=PS[K];
                  PS[K]:=PS[PVI];
                  PS[PVI]:=J;
                END;
             PV:=LU[PS[K],K];
             FOR I:=K+1 STEP 1 UNTIL N DO
                BEGIN
                  LU[PS[I],K]:=MULT:=LU[PS[I],K]/PV;
                  IF MULT NEQ 0 THEN
                     FOR J:=K+1 STEP 1 UNTIL N DO
                       LU[PS[I],J]:=LU[PS[I],J]-MULT*
                                    LU[PS[K],J];
                END;
     EL:  END;
          IF LU[PS[N],N]=0 THEN
               SINGULAR(1);
  END; %% DECOMPOSE %%
PROCEDURE SOLVE ;
  COMMENT  SOLVES AX=B USING LU FROM DECOMPOSE;
    BEGIN
       INTEGER  I,J;
       REAL     DOT;
            FOR I:=1 STEP 1 UNTIL N DO
              BEGIN
                 DOT:=0;
                 FOR J:=1 STEP 1 UNTIL I-1 DO
                      DOT:=DOT+LU[PS[I],J]*X[J];
                 X[I]:=B[PS[I]]-DOT;
              END;
            FOR I:=N STEP -1 UNTIL 1 DO
              BEGIN
                 DOT:=0;
                 FOR J:=I+1 STEP 1 UNTIL N DO
                      DOT:=DOT+LU[PS[I],J]*X[J];
                 X[I]:=(X[I]-DOT)/LU[PS[I],I];
              END;
    END; %% SOLVE %%
```

```
        %%%% MAIN %%%
        %
        %
             IF N=1 THEN
                  IF A[1,1] NEQ 0 THEN
                       X[1]:=B[1]/A[1,1]
                  ELSE
                       SINGULAR(1)
             ELSE
             IF N NEQ 0 THEN
               BEGIN
                  DECOMPOSE(N);
                  SOLVE;
              END;
        END; %% GAUSS-LU %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      PROCEDURE  USPE(N,P,Q,   MEAN,VAR,ATCR,  PHI,THETA,SIGMA,EPSILON);
          VALUE  N,P,Q,   MEAN,VAR,EPSILON;
          INTEGER  N,P,Q;
          REAL  MEAN,VAR,SIGMA,EPSILON;
          REAL ARRAY ATCR,THETA,PHI[*];
          % INPUT PARAMETERS
          %   N      :NUMBER OF W
          %   MEAN   :AVERAGE OF W
          %   VAR    :VARIANCE OF W
          %   ATCR   :AUTOCORRELATIONFUNCTION OF W
          %   P      :ASSUMED ORDER OF AUTO-REGRESSIVE MODEL
          %   PHI    :PARAMETERS FOR AUTO-REGRESSIVE  MODEL (TO BE ESTIMATED)
          %   Q      :ASSUMED ORDER OF MOVING-AVERAGE MODEL
          %   THETA  :PARAMETERS FOR MOVING-AVERAGE MODEL(TO BE ESTIMATED)
          %   SIGMA  :VARIANCE OF GAUSSIAN TERM (TO BE ESTIMATED)
          %   EPSILON:CRITERION FOR CONVERGENCE IN NEWTON-RAPHSON S METHOD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            BEGIN
         FORMAT OUT
          FF1(X10,"NO,OF DATA=",I3,X5,"MEAN=",R11.4,X5,"VARIANCE=",
             R11.4/),
          FF2(X10,"ORDER OF AR-PROCESS  ",I2,X10,"ORDER OF MA-PROCESS",
             X2,I2/),
          FF3(X10,"SIGMA=",R11.4,X5/),
          FF33(X10,"EPSILON=",R11.4/),
          FF4(X10,"AR-PROCESS PARAMETERS",X15,"MA-PROCESS PARAMETERS"/),
          FF5(X10,"PHI[",I2,"]",X1,R11.4,X20,"THETA[",I2,"]",X1,R11.4);
               REAL ARRAY    A[1:P,1:P], B[1:P],PHT[1:P],
                             T[1:Q+1,1:Q+1],TAU,C,F,H[1:Q+1];
               INTEGER   I,J,K,L,M;
               REAL      MAX;
               LABEL LOUT;
               %AR-PARAMETER ESTIMATED
               %
               FOR I:=1 STEP 1 UNTIL P DO
                 BEGIN
                    B[I]:=ATCR[Q+I]*VAR;
                    FOR  J:=1 STEP 1 UNTIL P DO
                         A[I,J]:=ATCR[ABS(Q+I-J)]*VAR;
                 END;
               %
               GAUSSLU(P,A,B,PHT);
               PHI[0]:=-1;
               FOR I:=1 STEP 1 UNTIL P DO
```

```
        PHI[I]:=PHT[I];
      %
      % MA-PROCESS  PARAMETER ESTIMATE
      %
      %(1) MODIFIED COVARIANCE SEQUENCE
      %
      FOR I:=1 STEP 1 UNTIL Q+1 DO
        BEGIN
          IF P EQL 0 THEN
               C[I]:=ATCR[I-1]*VAR
          ELSE
            BEGIN
               MAX:=0;
               FOR J:=0 STEP 1 UNTIL P DO
                    FOR K:=0 STEP 1 UNTIL P DO
                         MAX:=MAX+PHI[J]*PHI[K]*ATCR[ABS(J+I
                              -K-1)];
               C[I]:=MAX*VAR;
            END;
            END;  % I LOOP
      TAU[1]:=SQRT(C[1]);
      FOR I:=2 STEP 1 UNTIL Q+1 DO
          TAU[I]:=0;
      %NEWTON-RAPHSON METHOD
      FOR L:=1 STEP 1 UNTIL 10 DO
        BEGIN
          FOR I:=1 STEP 1 UNTIL Q+1 DO
            BEGIN
               FOR J:=1 STEP 1 UNTIL Q+1 DO
                 BEGIN
                    IF  I+J LEQ Q+1 THEN
                      BEGIN
                         IF J GEQ I THEN
                              T[I,J]:=TAU[I+J-1]+TAU[J-I+1]
                         ELSE
                              T[I,J]:=TAU[I+J-1];
                      END
                    ELSE
                    IF  J GEQ I THEN
                         T[I,J]:=TAU[J-I+1]
                    ELSE
                         T[I,J]:=0;
                 END; % J LOOP
               MAX:=0;
               FOR J:=1 STEP 1 UNTIL Q-I+2 DO
                    MAX:=MAX +TAU[J]*TAU[I+J-1];
               F[I]:=MAX-C[I];
            END;  %%%I LOOP
          GAUSSLU(Q+1,T,F,H);
          FOR I:=1 STEP 1 UNTIL Q+1 DO
               TAU[I]:=TAU[I]-H[I];
          MAX:=ABS(F[1]);
          FOR J:=2 STEP 1 UNTIL Q+1 DO
               IF MAX<ABS(F[J])  THEN
                    MAX:=ABS(F[J]);
          IF MAX LEQ EPSILON  THEN
               GO TO  LOUT;
        END;  % LOOP L
LOUT:
      FOR I:=1 STEP 1 UNTIL  Q DO
```

```
            THETA[I]:=-TAU[I+1]/TAU[1];
        IF P>0 THEN
          BEGIN
            MAX:=0;
            FOR I:=1 STEP 1 UNTIL P DO
                  MAX:=MAX+PHI[I];
           THETA[0]:=MEAN*(1-MAX);
          END   ELSE
            THETA[0]:=MEAN;
        IF Q>0 THEN
          BEGIN
            SIGMA:=TAU[1]*TAU[1];
          END
        ELSE
          BEGIN
            MAX:=0;
            FOR I:=1 STEP 1 UNTIL P DO
                  MAX:=MAX+PHI[I]*ATCR[I];
            SIGMA:=VAR*(1-MAX);
          END;
        WRITE(LINE[SKIP 1]);
        WRITE(LINE,<X10,"MESSAGES FROM USPE"//>);
        WRITE(LINE,FF1,N,MEAN,VAR);
        WRITE(LINE,FF2,P,Q);
        WRITE(LINE,FF3,SIGMA);
        WRITE(LINE,FF33,EPSILON);
        WRITE(LINE,FF4);
        IF P>Q THEN
          BEGIN
            FOR I:=0 STEP 1 UNTIL Q DO
                 WRITE(LINE,FF5,I,PHI[I],I,THETA[I]);
            FOR I:=Q+1    STEP 1 UNTIL P DO
                  WRITE(LINE,FF5,I,PHI[I],I,0);
          END   ELSE
          BEGIN
            FOR  I:=0 STEP 1 UNTIL P DO
                  WRITE(LINE,FF5,I,PHI[I],I,THETA[I]);
            FOR  I:=P+1    STEP 1 UNTIL  Q DO
                  WRITE(LINE,FF5,I,0,I,THETA[I]);
          END;
    END;  % USPE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   PROCEDURE   RESIDUAL(W,N,PHI,P,THETA,Q,MEAN,A);
      VALUE N,P,Q,MEAN;
      INTEGER    N,P,Q;
      REAL   MEAN;
      REAL ARRAY  PHI,THETA,W,A[*];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      BEGIN   COMMENT
            THIS SUBROUTINE COMPUTES THE RESIDUAL TERMS(GAUSSIAN
          TERMS) OF TIME SERIES BY BACK-FORECASTING AND FORWARD
          FORECASTING.THIS SUBROUTINE IS TO BE CALLED IN MLES(MAX
          IMUM LIKELIHOOD ESTIMATES) AND USNL(NON LINEAR APPROXI
          MATION);
        INTEGER I,J,K;
        REAL     TMP,TMQ;
        REAL ARRAY  E,X[-10-MAX(P,Q):N+Q-P];
             FOR I:=-10 STEP 1 UNTIL N DO
               BEGIN
                  A[I]:=0;
```

```
                IF I GEQ 1 AND I LEQ N THEN
                  BEGIN
                          X[I]:=W[I]-MEAN;
                  END
                ELSE
                     X[I]:=0;
                     E[I]:=0;
             END;
%         BACK-FORECASTING FOR E'S AND X'S.
%
          FOR I:=N-P STEP -1 UNTIL 1 DO
             BEGIN
                TMP:=TMQ:=0;
                FOR J:=1 STEP 1 UNTIL P DO
                     TMP:=TMP+PHI[J]*X[J+I];
                FOR J:=1 STEP 1 UNTIL Q DO
                     TMQ:=TMQ+THETA[J]*E[J+I];
                E[I]:=X[I]-TMP+TMQ;
             END;
          FOR I:=0 STEP -1 UNTIL -10 DO
             BEGIN
                TMP:=TMQ:=0;
                FOR J:=1 STEP 1 UNTIL P DO
                     TMP:=TMP+PHI[J]*X[J+I];
                FOR J:=1 STEP 1 UNTIL Q DO
                     TMQ:=TMQ+THETA[J]*E[J+I];
                X[I]:=E[I]+TMP-TMQ;
             END;
%         FORWARD FORECASTING FOR A'S
          FOR K:=-10 STEP 1 UNTIL N DO
             BEGIN
                TMP:=TMQ:=0;
                FOR J:=1 STEP 1 UNTIL P DO
                     TMP:=TMP+PHI[J]*X[K-J];
                FOR J:=1 STEP 1 UNTIL Q DO
                     TMQ:=TMQ+THETA[J]*A[-J+K];
                A[K]:=X[K]-TMP+TMQ;
             END;
    END OF RESIDUAL;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   REAL PROCEDURE   MLES(W,N,PHI,P,THETA,Q,MEAN,LS);
     VALUE      N,P,Q,MEAN;
     REAL  MEAN,LS;
    INTEGER   N,P,Q;
     REAL ARRAY  PHI,THETA,W[*];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
       BEGIN
            COMMENT    LEAST SQUARE FUNCTION IN CUMPUTED FOR
                   THE  GIVEN RESIDUAL TERMS;
         REAL ARRAY  A[-10-MAX(P,Q):N];
         REAL      TMP;
         INTEGER   I;
          RESIDUAL(W,N,PHI,P,THETA,Q,MEAN,A);
          %LEAST-SQUARE FUNCTION
          TMP:=0;
          FOR I:=-10 STEP 1 UNTIL N DO
               TMP:=TMP+A[I]*A[I];
          LS:=TMP;
       END;   %MLES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
      PROCEDURE  DGCK(A,N,PQ,T1,ITER);
         VALUE   N,PQ,T1,ITER;
         INTEGER   N,PQ,ITER;
         REAL   T1;
         REAL ARRAY  A[*];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            BEGIN
             REAL      MEAN,TMP,TO,CHISQ;
             INTEGER     I,J,K;
               TMP:=0;
               FOR I:=-10 STEP 1 UNTIL N DO
                    TMP:=TMP+A[I];
               MEAN:=TMP/N;
               CHISQ:=0;
               FOR K:=0 STEP 1 UNTIL 20 DO
                 BEGIN
                    TMP:=0;
                    FOR I:=-10 STEP 1 UNTIL N-K DO
                         TMP:=TMP+(A[I]-MEAN)*(A[I+K]-MEAN);
                    IF K NEQ 0 THEN
                         CHISQ:=CHISQ+(TMP/TO)**2
                    ELSE
                         TO:=TMP;
                 END OF K LOOP;
               CHISQ:=CHISQ*N;
               WRITE(LINE,<X4,I3,X59,R11.4,X2,R11.4,X7,I3 >,ITER,T1,
                         CHISQ,K-PQ-1);
         END OF DGCK;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      PROCEDURE    MLPRINT(Z,N,PHI,P,THETA,Q,MEAN,DLT);
         VALUE     N,P,Q,MEAN,DLT;
         INTEGER      N,P,Q;
         REAL     MEAN,DLT;
         REAL  ARRAY    Z,PHI,THETA[*];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
             %
             %
             %  N      NUMBER OF DATA
             %  Z      TIME SERIES
             %  PHI    AUTO-REGRESSIVE PARAMETERS
             %  P      THE ORDER OF AR PARAMETERS
             %  THETA   MOVING AVERAGING PARAMETERS
             %  Q       THE ORDER OF MA PARAMETERS
             %  MEAN    THE MEAN VALUE OF Z
             %  DLT     SMALL CHANGES OF PARAMETERS
             %
            %       THIS PROCEDURE PRINTS OUT THE SUM OF THE SQUARES FUNCTION
            %   WHEN THERE ARE 1 OR 2 PARAMETERS.
            %
            %   1 DIM.      21 VALUES AROUND THE PRE-ESTIMATED PARAMETER
            %   2 DIM.      11*11 VALUES AROUND THE PRE-ESTIMATED PARAMETERS.
            %
           BEGIN
             FORMAT OUT    FM1(11(X2,R9.2)//),
                           FM2(11(X2,R9.2)//),
                           FM4(R9.2,"*",11(X2,R9.2)/,X9,"*"/,X9,"*"),
                          FM3(X10,11(X2,R9.2));
             REAL ARRAY    CTHETA[0:Q],CPHI[0:P],LS[-5:5];
             REAL   A,B,PA;
             INTEGER   I,J,K;
```

```
%
 WRITE(LINE[SKIP 1]);
      WRITE(LINE,<X10,"MESSAGES FROM MLPMINT  (THE SUM"
            ," OF THE SQUARE FUNCTION"//>);
 IF P+Q=1 THEN
   BEGIN
      FOR J:=-5 STEP 10 UNTIL 5 DO
      BEGIN
      FOR I:=-5 STEP 1 UNTIL 5 DO
        BEGIN
           IF P=1 THEN
                CPHI[1]:= PHI[1]+DLT*(I+J)
           ELSE
                CTHETA[1]:= THETA[1]+DLT*(I+J);
           MLES(Z,N,CPHI,P,CTHETA,Q,MEAN,LS[I]);
        END;
           IF P=0 THEN
                PA:=THETA[1]
           ELSE
                PA:=PHI[1];
      WRITE(LINE,FM1,FOR I:=-5 STEP 1 UNTIL 5 DO
            PA+(I+J)*DLT);
      WRITE(LINE,FM2,FOR I:=-5 STEP 1 UNTIL 5 DO
            LS[I]);
      WRITE(LINE[SPACE 2]);
      END;
   END  ELSE
   BEGIN
      IF P=0 THEN
        BEGIN
           A:=THETA[1];
           B:=THETA[2];
           WRITE(LINE,</X20,"A=THETA[1]",X5,"B=THETA[2]"
                //>);
        END  ELSE
      IF P=1 THEN
        BEGIN
           A:=PHI[1];
           B:=THETA[1];
           WRITE(LINE,</X20,"A=PHI[1]",X7,"B=THETA[1]"//>
                );
        END ELSE
        BEGIN
           A:=PHI[1];
           B:=PHI[2];
           WRITE(LINE,</X20,"A=PHI[1]",X7,"B=PHI[2]"//>);
        END;
      WRITE(LINE,<X115,"A"/>);
      WRITE(LINE,FM3,FOR I:=-5 STEP 1 UNTIL 5
            DO A+I*DLT);
      WRITE(LINE,</X9,121("*")>);
      FOR I:=-5 STEP 1 UNTIL 5 DO
        BEGIN
           IF P=0 THEN
                CTHETA[2]:= THETA[2]+DLT*I
           ELSE IF P=1 THEN
                CTHETA[1]:=THETA[1]+I*DLT
           ELSE
                CPHI[2]:= PHI[2]+DLT*I;
                FOR J:=-5 STEP 1 UNTIL 5 DO
```

```
                                BEGIN
                                   IF P=0 THEN
                                        CTHETA[1]:= THETA[1]+DLT*J
                                   ELSE
                                        CPHI[1]:=PHI[1]+J*DLT;
                                   MLES(Z,N,CPHI,P,CTHETA,Q,MEAN,LS[J])
                                        ;
                                END;
                              WRITE(LINE,FM4,B+I*DLT,FOR J:=-5 STEP 1
                                   UNTIL 5 DO  LS[J]);
                         END;
                       WRITE(LINE,</X3,"B">);
                   END;
                       WRITE(LINE[SKIP 1]);
              END;  %% SUM OF THE SQUARES PRINT %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      PROCEDURE  MLQT(W,N,PHI,P,THETA,Q,MEAN,EPSILON,DELTA,IM,FACTOR,
                   LAMBDA,LMAX);
         VALUE   N,P,Q,MEAN,EPSILON,DELTA,IM,FACTOR,LAMBDA,LMAX;
         REAL   MEAN,EPSILON,DELTA,FACTOR,LAMBDA,LMAX;
         INTEGER   N,P,Q,IM;
         REAL ARRAY  PHI,THETA,W[*];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %
            %
            %
            %W        TIME SERIES
            %N        NUMBER OF DATA(W)
            %PHI     AR PARAMETERS
            %P        ORDER OF PHI
            %THETA   MA PARAMETERS
            %Q        ORDER OF THETA
            %MEAN     AVERAGE OF W
            %EPSILON MAXIMUM ERROR OF CONVERGENCE OF PARAMETERS
            %DLT      DELTA USED TO CALCULATE THE DERIVATIVES
            %IM      MAXIMUM NUMBER OF ITERATION
            %FACTOR PARAMETER TO MODIFY LAMBDA DEPENDING ON THE CONVERGENCE
            %LAMBDA PARAMETER TO MODIFY THE DIAGONAL ELEMENTS IN MARQUARDT'S
            %         ALGORITHM
            %LMAX   MAXIMUM OFLAMBDA
            %
            %
            %     THIS PROCEDURE CALCULATES THE PARAMETERS ITERATIVELY.
            %CONVERGENCE IS BETTER THAN GAUSS-NEWTON'S METHOD(USNL).
            %THE PROGRAM IS ALMOST IDENTICAL TO USNL EXCEPT THAT
            %DIAGONAL ELEMENTS ARE MODIFIED TO GIVE BETTER CONVERGENCE.
           %THE DERIVATIVES ARE CALCULATED AT FIRST.
         BEGIN
         FORMAT   OUT
         FF4(X105,2(X2,R9,2)/),
         FF6(" # OF ITER,",X9,"PHI",X20,"THETA",X18,"SUM OF SQUARES",
             X2,"CHI-SQUARE",X2,"DEG. OF FREE.",X2,"LAMBDA",
             X4,"MAXBETA"//),
         FF5(X12,"PHI[",I2,"]",X1,R11,4,X04,"THETA[",I2,"]",X1,R11,4);
           REAL  TMP,DELTAINV,MAC,LS,LSO;
           INTEGER    I,J,K,PQ,ITER;
          REAL ARRAY A,A1[-10-Q:N+P+Q],X[1:P+Q,-10:N+P+Q],B[1:P+Q,
                    1:P+Q],F,BETA[1:P+Q];
           LABEL     L2,LSICK,LOUT;
           REAL ARRAY     D[1:P+Q];
```

```
MLES(W,N,PHI,P,THETA,Q,MEAN,LSO);
WRITE(LINE[SKIP 1]);
WRITE(LINE,<X15,"MESSAGES FROM MLQT"//>);
WRITE(LINE,FF6);
DELTAINV:=1/DELTA;
PQ:=P+Q;
RESIDUAL(W,N,PHI,P,THETA,Q,MEAN,A);
FOR  ITER:=1 STEP 1 UNTIL  IM DO
  BEGIN
     FOR I:=1 STEP 1 UNTIL PQ DO
       BEGIN
          IF I LEQ P THEN
               PHI[I]:=PHI[I]+DELTA
          ELSE
               THETA[I-P]:=THETA[I-P]+DELTA;
          RESIDUAL(W,N,PHI,P,THETA,Q,MEAN,A1);
          IF I LEQ P THEN
               PHI[I]:=PHI[I]-DELTA
          ELSE
               THETA[I-P]:=THETA[I-P]-DELTA;
           FOR J:=-10 STEP 1 UNTIL N DO
               X[I,J]:=(A[J]-A1[J])*DELTAINV;
       END;  % OF COMPUTING DERIVATIVES
          FOR I:=1 STEP 1 UNTIL PQ DO
            BEGIN
               FOR J:=I STEP 1 UNTIL PQ DO
                 BEGIN
                    TMP:=0;
                    FOR K:=-10 STEP 1 UNTIL N DO
                         TMP:=TMP+X[I,K]*X[J,K];
                    B[I,J]:=TMP;
                    IF I NEQ J THEN
                         B[J,I]:=TMP;
                 END; % OF J LOOP
               D[I]:=SQRT(B[I,I]);
               TMP:=0;
               FOR K:=-10 STEP 1 UNTIL N DO
                    TMP:=TMP+X[I,K]*A[K];
               F[I]:=TMP/D[I];
            END OF I LOOP;
          FOR I:=1 STEP 1 UNTIL PQ DO
          FOR J:=I STEP 1 UNTIL PQ DO
               IF I NEQ J THEN
                    B[I,J]:=B[J,I]:=B[I,J]/(D[I]*D[J]);
 L2:
          FOR I:=1 STEP 1 UNTIL PQ DO
               B[I,I]:=1+LAMBDA;
          GAUSSLU(PQ,B,F,BETA);
          FOR I:=1 STEP 1 UNTIL PQ DO
               BETA[I]:=BETA[I]/D[I];
          FOR I:=1 STEP 1 UNTIL PQ DO
               IF I LEQ P THEN
                    PHI[I]:=PHI[I]+BETA[I]
               ELSE
               THETA[I-P]:=THETA[I-P]+BETA[I];
          MLES(W,N,PHI,P,THETA,Q,MEAN,LS);
          IF LS < LSO THEN
            BEGIN
               LSO:=LS;
RESIDUAL(W,N,PHI,P,THETA,Q,MEAN,A);
```

```
TMP:=0;
FOR K:=-10 STEP 1 UNTIL N DO
  BEGIN
     TMP:=TMP+A[K]*A[K];
  END;
     DGCK(A,N,P+Q,TMP,ITER);
IF P>Q THEN
  BEGIN
     FOR I:=0 STEP 1 UNTIL Q DO
         WRITE(LINE,FF5,I,PHI[I],I,THETA[I]);
     FOR I:=Q+1   STEP 1 UNTIL P DO
          WRITE(LINE,FF5,I,PHI[I],I,0);
  END   ELSE
  BEGIN
     FOR  I:=1 STEP 1 UNTIL P DO
          WRITE(LINE,FF5,I,PHI[I],I,THETA[I]);
     FOR  I:=P+1   STEP 1 UNTIL  Q DO
          WRITE(LINE,FF5,I,0,I,THETA[I]);
  END;
          MAC:=ABS(BETA[1]);
          FOR I:=2 STEP 1 UNTIL PQ DO
               MAC:=MAX(MAC,ABS(BETA[I]));
          IF MAC<EPSILON THEN GO TO LOU;
          WRITE(LINE,FF4,LAMBDA,MAC);
          LAMBDA:=LAMBDA/FACTOR;
            END   ELSE
            BEGIN
               FOR I:=1 STEP 1 UNTIL PQ DO
                    IF I LEQ P THEN
                         PHI[I]:=PHI[I]-BETA[I]
                    ELSE
                         THETA[I]:=THETA[I]-BETA[I];
               LAMBDA:=LAMBDA*FACTOR;
               IF LAMBDA>LMAX THEN GO TO LSICK;
               GO TO L2;
            END;
  END;
  LSICK:
     IF P>0 THEN
       BEGIN
          TMP:=0;
          FOR I:=1 STEP 1 UNTIL P DO
               TMP:=TMP+PHI[I];
          THETA[0]:=MEAN*(1-TMP);
       END ELSE
     THETA[0]:=MEAN;
     WRITE(LINE,</X35,"THETA[0]=",R11,4///>,THETA[0]);
     WRITE(LINE,<X123,"*">);
     WRITE(LINE,<X123,"*">);
     WRITE(LINE,<X10,"NO MORE CONVERGENCE.CHECK IF ",
         "SOLUTIONS ARE REASONABLE OR NOT,",52("*")>);
     WRITE(LINE,<X5,"IF MAXBETA IS VERY SMALL OR THE ",
     "CONVERGENCE OF PARAMETERS IS OBVIOUSLY GOOD,",
     "YOU ARE GETTING RIGHT ANSWERS.  ON THE OTHER ",
     "HAND,IF MAXBETA IS STILL LARGE AFTER MAX ",
     "ITERATION OR IF THE SUM OF SQUARES IS NOT ",
     "APPROACHING THE MINIMUM VALUE, YOU SHOULD ",
     "RE-EXAMINE THE DATA. THIS MESSAGE COMES WHEN ",
     "LAMBDA REACHES OR EXCEEDS ",I4/>,LMAX);
LOUT:
```

```
        END OF MLQT;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%                              %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%      MAIN PROGRAM            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%                              %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        READ(CARD,/,N,P,D,Q,IM,DLT,EPSILON,GAMA,ZETA);
        BEGIN
          REAL ARRAY W,Z[1:N+10],A[-15:N+10];
          LABEL    FINISH;
            FOR I:=0 STEP 1 UNTIL N DIV 7 DO
            READ(CARD,/,FOR J:=1 STEP 1 UNTIL 7 DO Z[7*I+J])[FINISH];
        FINISH:
            WRITE(LINE,<X10,"INPUT DATA (TIME SERIES)"///>);
            FOR I:=0 STEP 1 UNTIL N DIV 7 DO
                  WRITE(LINE,<X2,7(F8.5,X3)>,FOR J:=1 STEP 1 UNTIL 7
                        DO Z[I*7+J]);
            USID(Z,N,MEAN,ATCR,VAR,20,20,D,W);
            WRITE(LINE[SKIP 1]);
            WRITE(LINE,<X10,"MODIFIED TIME SERIES"///>);
            FOR I:=0 STEP 1 UNTIL N DIV 7 DO
                  WRITE(LINE,<X2,7(F8.5,X3)>,FOR J:=1 STEP 1 UNTIL 7
                        DO W[I*7+J]);
            USPE(N-D,P,Q,MEAN,VAR,ATCR,  PHI,THETA,SIGMA,EPSILON);
            MLPRINT(W,N-D,PHI,P,THETA,Q,MEAN,DLT);
          MLQT(W,N-D,PHI,P,THETA,Q,MEAN,EPSILON,DLT,20,2.0,.01,100);
          END;
    END.
```

APPENDIX 2

ARMA (1, 0, 1) Model

INPUT DATA (TIME SERIES)

17.00000	16.60000	16.30000	16.10000	17.10000	16.90000	16.80000
17.40000	17.10000	17.00000	16.70000	17.40000	17.20000	17.40000
17.40000	17.00000	17.30000	17.20000	17.40000	16.80000	17.10000
17.40000	17.40000	17.50000	17.40000	17.60000	17.40000	17.30000
17.00000	17.80000	17.50000	18.10000	17.50000	17.40000	17.40000
17.10000	17.60000	17.70000	17.40000	17.80000	17.60000	17.50000
16.50000	17.80000	17.30000	17.30000	17.10000	17.40000	16.90000
17.30000	17.60000	16.90000	16.70000	16.80000	16.80000	17.20000
16.80000	17.60000	17.20000	16.60000	17.10000	16.90000	16.60000
18.00000	17.20000	17.30000	17.00000	16.90000	17.30000	16.80000
17.30000	17.40000	17.70000	16.80000	16.90000	17.00000	16.90000
17.00000	16.60000	16.70000	16.80000	16.70000	16.40000	16.50000
16.40000	16.60000	16.50000	16.70000	16.40000	16.40000	16.20000
16.40000	16.30000	16.40000	17.00000	16.90000	17.10000	17.10000
16.70000	16.90000	16.50000	17.20000	16.40000	17.00000	17.00000
16.70000	16.20000	16.60000	16.90000	16.50000	16.60000	16.60000
17.00000	17.10000	17.10000	16.70000	16.80000	16.30000	16.60000
16.80000	16.90000	17.10000	16.80000	17.00000	17.20000	17.30000
17.20000	17.30000	17.20000	17.20000	17.50000	16.90000	16.90000
16.90000	17.00000	16.50000	16.70000	16.80000	16.70000	16.70000
16.60000	16.50000	17.00000	16.70000	16.70000	16.90000	17.40000
17.10000	17.00000	16.80000	17.20000	17.20000	17.40000	17.20000
16.90000	16.80000	17.00000	17.40000	17.20000	17.20000	17.10000
17.10000	17.10000	17.40000	17.20000	16.90000	16.90000	17.00000
16.70000	16.90000	17.30000	17.80000	17.80000	17.60000	17.50000
17.00000	16.90000	17.10000	17.20000	17.40000	17.50000	17.90000
17.00000	17.00000	17.00000	17.20000	17.30000	17.40000	17.40000
17.00000	18.00000	18.20000	17.60000	17.80000	17.70000	17.20000
17.40000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

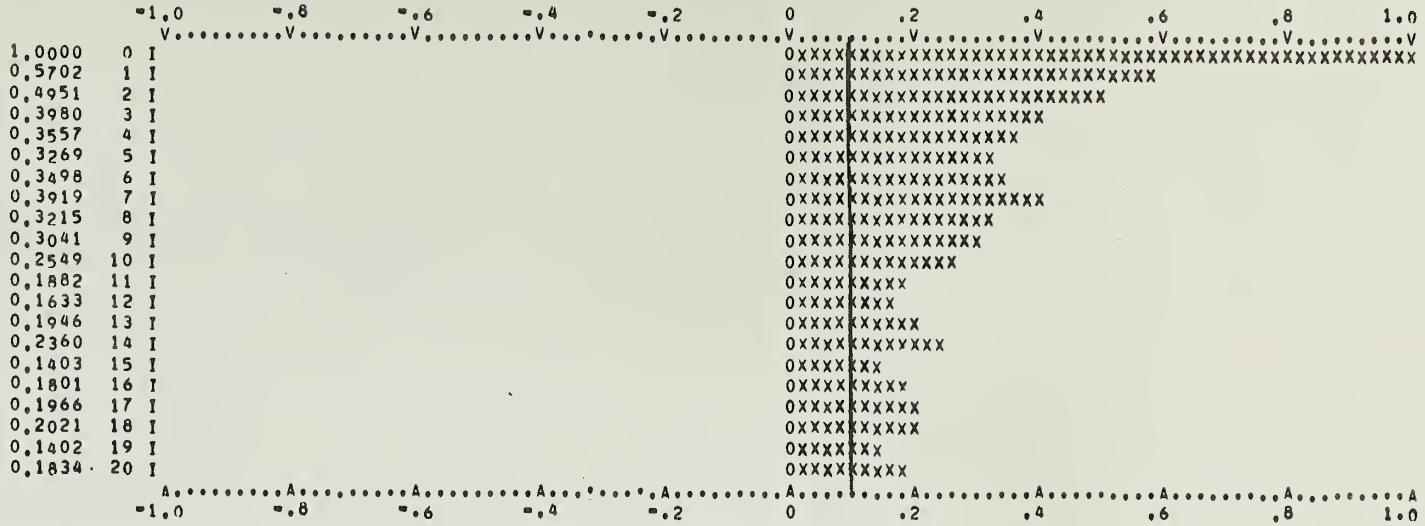
MESSAGES FROM USID

NUMBER OF DATA= 197
ORDER OF THE DIFFERENCE = 0

MEAN VALUE = 17.0624
VARIANCE = 0.1586

ORDER	AUTOCORRELATIONS	PARTIAL AUTOCORRELATIONS
0	1.0000	0.0000
1	0.5702	0.5702
2	0.4951	0.2518
3	0.3980	0.0683
4	0.3557	0.0693
5	0.3269	0.0658
6	0.3498	0.1237
7	0.3919	0.1563
8	0.3215	-0.0317
9	0.3041	0.0098
10	0.2549	-0.0164
11	0.1882	-0.0726
12	0.1633	-0.0214
13	0.1946	0.0606
14	0.2360	0.0872
15	0.1403	-0.1235
16	0.1801	0.0472
17	0.1966	0.0969
18	0.2021	0.0681
19	0.1402	-0.0713
20	0.1834	0.0522

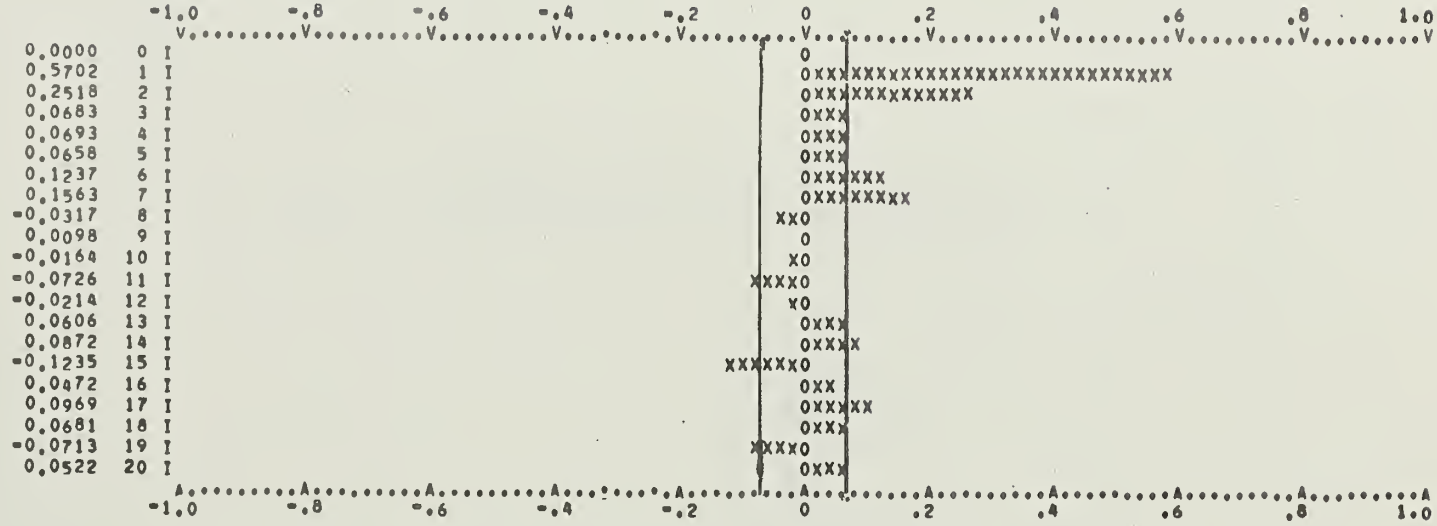
AUTO CORRELATION FUNCTION



IF THE PROCESS IS MA(Q), THEN THE AUTOCORRELATIONS HIGHER THAN Q ARE WITHIN THE RANGE OF:

Q	ERROR BOUND
1	0.0915
2	0.1042
3	0.1117
4	0.1173
5	0.1218
6	0.1268
7	0.1328
8	0.1367
9	0.1401
10	0.1424
11	0.1437
12	0.1446
13	0.1460
14	0.1479
15	0.1486
16	0.1497
17	0.1510
18	0.1523
19	0.1530
20	0.1541

PARTIAL AUTOCORRELATION FUNCTION

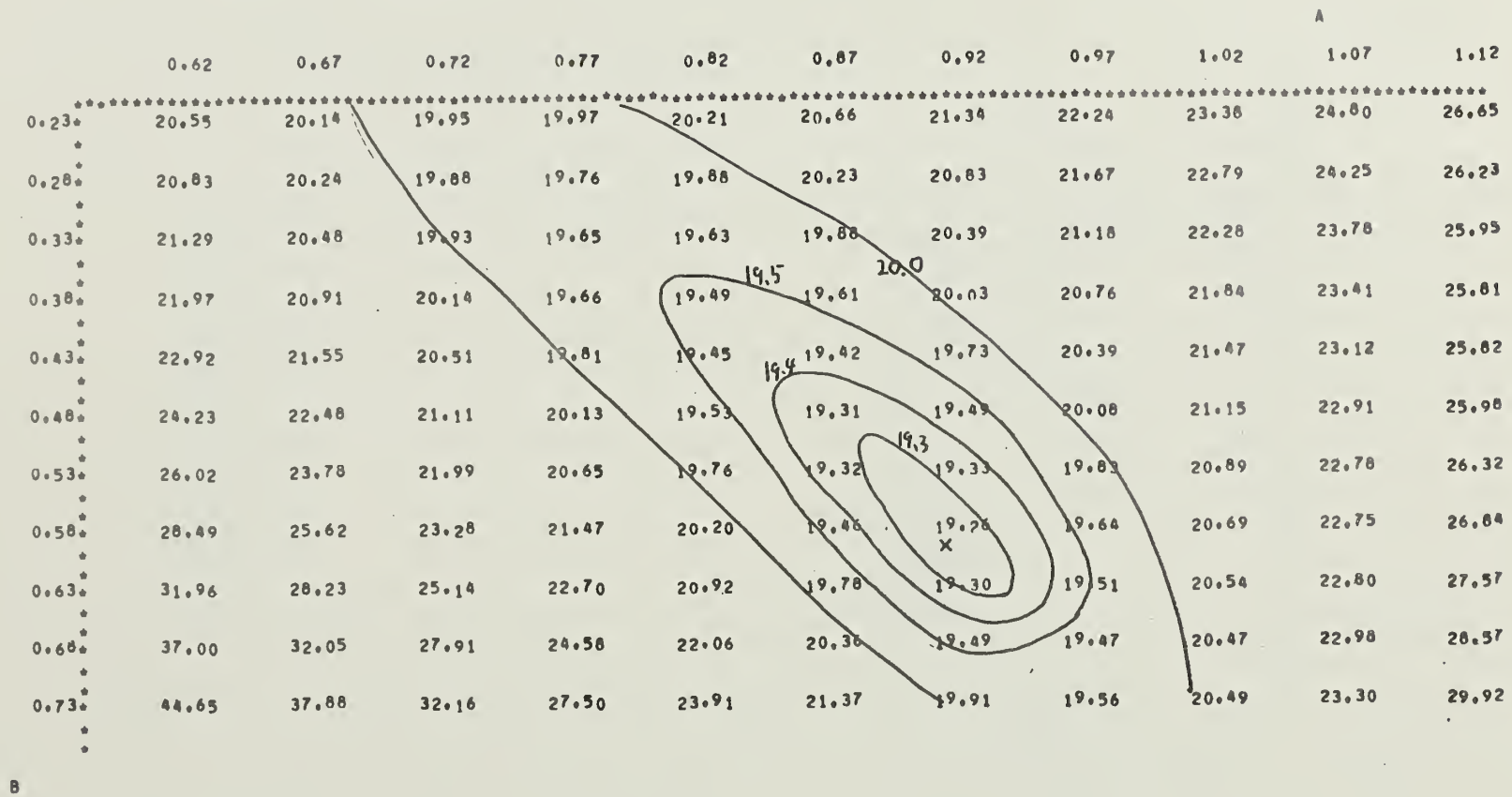


IF THE PROCESS IS AR(P), THE PARTIAL AUTOCORRELATIONS HIGHER THAN P ARE WITHIN THE RANGE OF: 0.0712

```
MESSAGES FROM USPE

NO.OF DATA=197      MEAN=     17.0624      VARIANCE=      0.1586

ORDER OF AR-PROCESS   1           ORDER OF MA-PROCESS   1

SIGMA=      0.0988

EPSILON=      0.0010

AR-PROCESS PARAMETERS                MA-PROCESS PARAMETERS

PHI( 0)      -1.0000                    THETA( 0)       2.2475
PHI( 1)       0.8683                    THETA( 1)       0.4773
```

MESSAGES FROM MLPRINT (THE SUM OF THE SQUARE FUNCTION

A=PHI[1] B=THETA[1]



MESSAGES FROM MLQT

# OF ITER.		PHI		THETA		SUM OF SQUARES	CHI-SQUARE	DEG. OF FREE.	LAMBDA	MAXBETA
1						19.2603	22.9809	18		
	PHI[1]	0.8996	THETA[1]	0.5427						
									1.00E-02	0.07
2						19.2541	23.0928	18		
	PHI[1]	0.9102	THETA[1]	0.5715						
									5.00E-03	0.03
			THETA[0]=	1.5329						

NO MORE CONVERGENCE.CHECK IF SOLUTIONS ARE REASONABLE OR NOT.**********
IF MAXBETA IS VERY SMALL OR THE CONVERGENCE OF PARAMETERS IS OBVIOUSLY GOOD,YOU ARE GETTING RIGHT ANSWERS. ON THE OTHER HAND.IF MAXBETA IS STILL LARGE AFTER MAX ITERATION OR IF THE SUM OF SQUARES IS NOT APPROACHING THE MINIMUM VALUE. YOU SHOULD RE-EXAMINE THE DATA. THIS MESSAGE COMES WHEN LAMBDA REACHES OR EXCEEDS 100

$$Z_t - .91 Z_{t-1} = 1.53 + a_t - .57 a_{t-1}$$

APPENDIX 3

(ARMA (1, 0, 1) Model

MODIFIED TIME SERIES

-0.40000	-0.30000	-0.20000	1.00000	-0.20000	-0.10000	0.60000
-0.30000	-0.10000	-0.30000	0.70000	-0.20000	0.20000	0.00000
-0.40000	0.30000	-0.10000	0.20000	-0.60000	0.30000	0.30000
0.00000	0.10000	-0.10000	0.20000	-0.20000	-0.10000	-0.30000
0.80000	-0.30000	0.60000	-0.60000	-0.10000	0.00000	-0.30000
0.50000	0.10000	-0.30000	0.40000	-0.20000	-0.10000	-1.00000
1.30000	-0.50000	0.00000	-0.20000	0.30000	-0.50000	0.40000
0.30000	-0.70000	-0.20000	0.10000	0.00000	0.40000	-0.40000
0.80000	-0.40000	-0.60000	0.50000	-0.20000	-0.30000	1.40000
-0.80000	0.10000	-0.30000	-0.10000	0.40000	-0.50000	0.50000
0.10000	0.30000	-0.90000	0.10000	0.10000	-0.10000	0.10000
-0.40000	0.10000	0.10000	-0.10000	-0.30000	0.10000	-0.10000
0.20000	-0.10000	0.20000	-0.30000	0.00000	-0.20000	0.20000
-0.10000	0.10000	0.60000	-0.10000	0.20000	0.00000	-0.40000
0.20000	-0.40000	0.70000	-0.80000	0.60000	0.00000	-0.30000
-0.50000	0.40000	0.30000	-0.40000	0.10000	0.00000	0.40000
0.10000	0.00000	-0.40000	0.10000	-0.50000	0.30000	0.20000
0.10000	0.20000	-0.30000	0.20000	0.20000	0.10000	-0.10000
0.10000	-0.10000	0.00000	0.30000	-0.60000	0.00000	0.00000
0.10000	-0.50000	0.20000	0.10000	-0.10000	0.00000	-0.10000
-0.10000	0.50000	-0.30000	0.00000	0.20000	0.50000	-0.30000
-0.10000	-0.20000	0.40000	0.00000	0.20000	-0.20000	-0.30000
-0.10000	0.20000	0.40000	-0.20000	0.00000	-0.10000	0.00000
0.00000	0.30000	-0.20000	-0.30000	0.00000	0.10000	-0.30000
0.20000	0.40000	0.50000	0.00000	-0.20000	-0.10000	-0.50000
-0.10000	0.20000	0.10000	0.20000	0.10000	0.40000	-0.90000
0.00000	0.00000	0.20000	0.10000	0.10000	0.00000	-0.40000
1.00000	0.20000	-0.60000	0.20000	-0.10000	-0.50000	0.20000
17.40000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

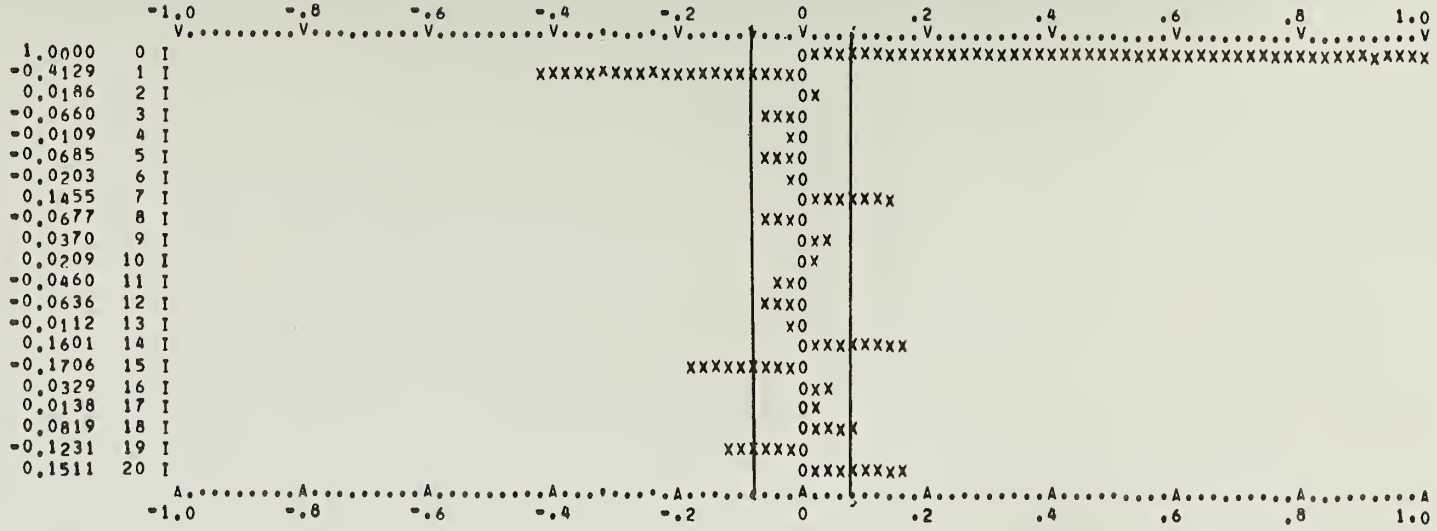
MESSAGES FROM USID

NUMBER OF DATA= 197
ORDER OF THE DIFFERENCE = 1

MEAN VALUE = 0.0020
VARIANCE = 0.1364

ORDER	AUTOCORRELATIONS	PARTIAL AUTOCORRELATIONS
0	1.0000	0.0000
1	-0.4129	-0.4129
2	0.0186	-0.1831
3	-0.0660	-0.1653
4	-0.0109	-0.1391
5	-0.0685	-0.1926
6	-0.0203	-0.2139
7	0.1455	-0.0021
8	-0.0677	-0.0466
9	0.0370	-0.0184
10	0.0209	0.0415
11	-0.0460	-0.0054
12	-0.0636	-0.0766
13	-0.0112	-0.1019
14	0.1601	0.1031
15	-0.1706	-0.0851
16	0.0329	-0.1270
17	0.0138	-0.0959
18	0.0819	0.0454
19	-0.1231	-0.0737
20	0.1511	0.0911

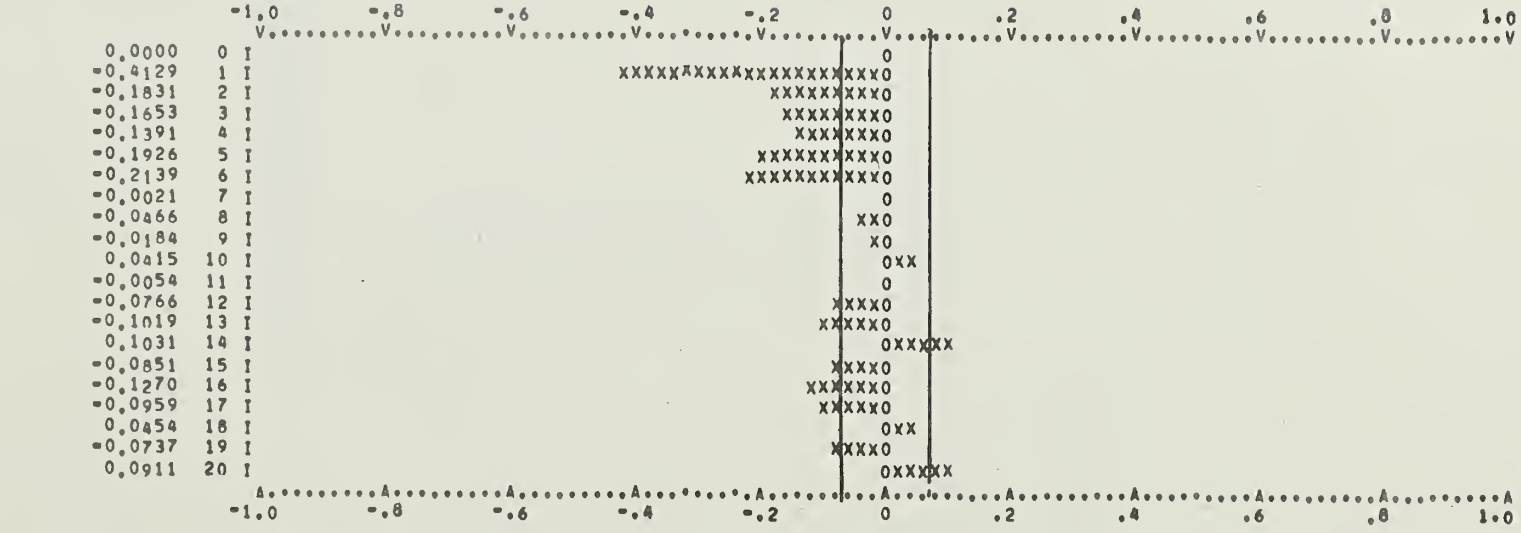
AUTO CORRELATION FUNCTION



IF THE PROCESS IS MA(Q), THEN THE AUTOCORRELATIONS HIGHER THAN Q ARE WITHIN THE RANGE OF:

Q	ERROR BOUND
1	0.0827
2	0.0827
3	0.0830
4	0.0830
5	0.0833
6	0.0833
7	0.0846
8	0.0849
9	0.0850
10	0.0850
11	0.0851
12	0.0854
13	0.0854
14	0.0869
15	0.0886
16	0.0886
17	0.0887
18	0.0890
19	0.0899
20	0.0912

PARTIAL AUTOCORRELATION FUNCTION

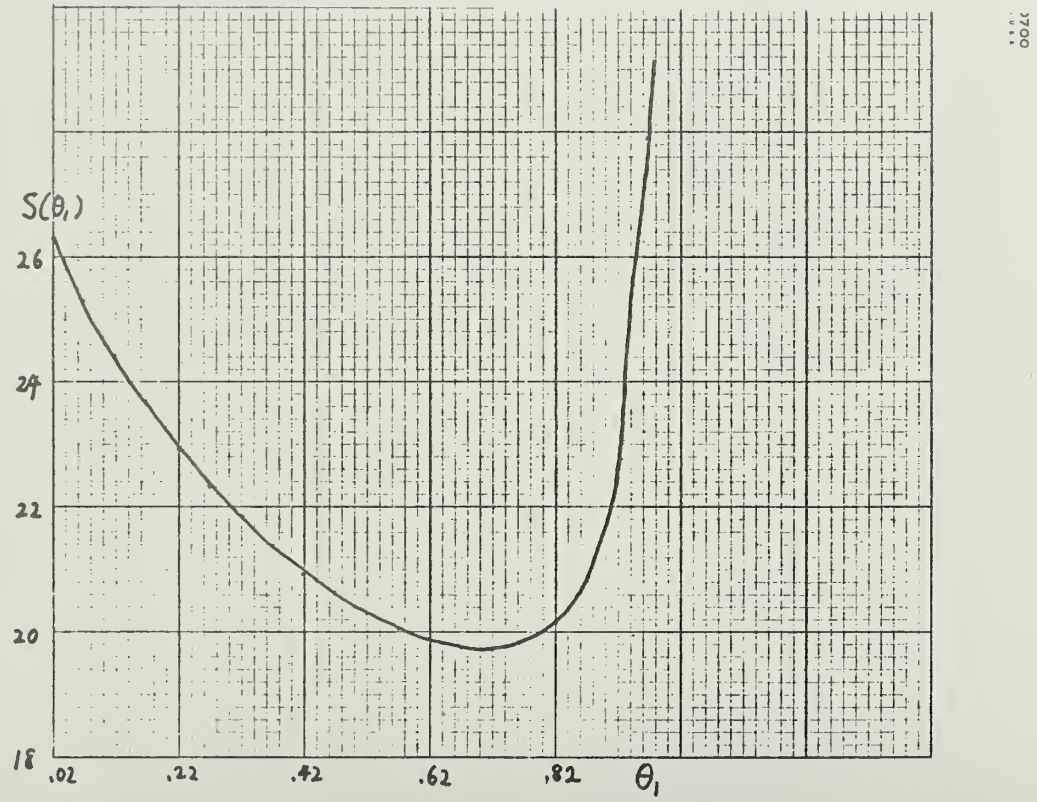


IF THE PROCESS IS AR(P), THE PARTIAL AUTOCORRELATIONS HIGHER THAN P ARE WITHIN THE RANGE OF: 0.0714

```
MESSAGES FROM USPE

NO.OF DATA=196      MEAN=      0.0020      VARIANCE=      0.1364

ORDER OF AR-PROCESS   0          ORDER OF MA-PROCESS   1

SIGMA=      0.1075

EPSILON=      0.0010

AR-PROCESS PARAMETERS                MA-PROCESS PARAMETERS

PHI[ 0]      -1.0000                    THETA[ 0]      0.0020
PHI[ 1]       0.0000                    THETA[ 1]      0.5223
```

MESSAGES FROM MLPRINT (THE SUM OF THE SQUARE FUNCTION

0.02	0.07	0.12	0.17	0.22	0.27	0.32	0.37	0.42	0.47	0.52
26.26	25.28	24.41	23.64	22.96	22.36	21.83	21.37	20.96	20.61	20.31
0.52	0.57	0.62	0.67	0.72	0.77	0.82	0.87	0.92	0.97	1.02
20.31	20.06	19.87	19.75	19.72	19.83	20.17	20.92	22.64	27.96	1.44E 07



MESSAGES FROM MLQT

# OF ITER.	PHI		THETA		SUM OF SQUARES	CHI-SQUARE	DEG. OF FREE.	LAMBDA	MAXBETA
1					19.7978	27.2768	19		
	PHI[1]	0.0000	THETA[1]	0.6405				1.00E-02	0.12
2					19.7199	26.6599	19		
	PHI[1]	0.0000	THETA[1]	0.6946				5.00E-03	0.05
3					19.7154	26.4984	19		
	PHI[1]	0.0000	THETA[1]	0.7124				2.50E-03	0.02
			THETA[0]=	0.0020					

NO MORE CONVERGENCE.CHECK IF SOLUTIONS ARE REASONABLE OR NOT.**********

IF MAXBETA IS VERY SMALL OR THE CONVERGENCE OF PARAMETERS IS OBVIOUSLY GOOD.YOU ARE GETTING RIGHT ANSWERS. ON THE OTHER HAND.IF MAXBETA IS STILL LARGE AFTER MAX ITERATION OR IF THE SUM OF SQUARES IS NOT APPROACHING THE MINIMUM VALUE, YOU SHOULD RE-EXAMINE THE DATA. THIS MESSAGE COMES WHEN LAMBDA REACHES OR EXCEEDS 100

$$Z_t - Z_{t-1} = .002 + a_t - .71 a_{t-1}$$

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Center for Advanced Computation University of Illinois at Urbana-Champaign Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

REPORT TITLE
An Introduction to the Analysis of Time Series

DESCRIPTIVE NOTES (Type of report and inclusive dates)
Research Report

AUTHOR(S) (First name, middle initial, last name)
K. Miura

REPORT DATE November 8, 1971	7a. TOTAL NO. OF PAGES 86	7b. NO. OF REFS 10
a. CONTRACT OR GRANT NO. DAHC04-72-C-0001 b. PROJECT NO. ARPA Order 1899 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) CAC Document No. 26 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

0. DISTRIBUTION STATEMENT
Copies may be obtained from the address quoted in (1) above.

1. SUPPLEMENTARY NOTES NONE	12. SPONSORING MILITARY ACTIVITY U.S. Army Research Office-Durham Duke Station Durham, North Carolina

3. ABSTRACT

This document is an introduction to the analysis of time series using autocorrelations. The autoregressive, the moving average, and the mixed models for stationary and nonstationary time series--are discussed. Several stages to analyze given time series are discussed. These are: identification, preestimation of parameters, the maximum likelihood estimation and diagnostic checking.

Numerical techniques are described in detail together with some examples.

DD FORM 1 NOV 68 1473

UNCLASSIFIED
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Mathematical Statistics; Probability						